(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 954 181 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
03.11.1999 Bulletin 1999/44

(51) Int. Cl.6: H04N 7/24

(21) Application number: 99303221.8

(22) Date of filing: 26.04.1999

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 28.04.1998 JP 11873798
16.11.1998 JP 32563598
16.11.1998 JP 32563698

(71) Applicant:
CANON KABUSHIKI KAISHA
Tokyo (JP)

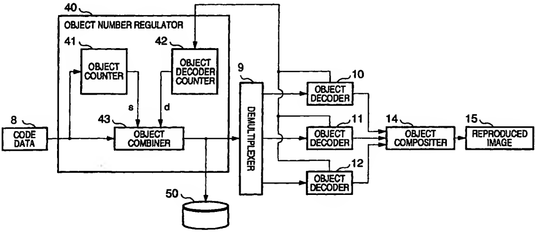
(72) Inventor: Maeda, Mitsuru
Ohta-ku, Tokyo (JP)

(74) Representative:
Beresford, Keith Denis Lewis et al
BERESFORD & Co.
High Holborn
2-5 Warwick Court
London WC1R 5DJ (GB)

(54) Data processing apparatus and method

(57) In decoding code data encoded in object units, decoders corresponding to the number of objects are needed. However, it is impossible to always provide a sufficient number of decoder. Accordingly, when code data 8 is decoded, an object combiner 43 refers to the number s of objects included in the code data 8, detected by an object counter 41, and the number d of object decoders, detected by an object decoder counter 42. If s>d holds, the object combiner 43 regulates the number of the objects of the input code data 8 to d.

FIG. 7



EP 0 954 181 A2

Description

BACKGROUND OF THE INVENTION

FIELD OF THE INVENTION

[0001] The present invention relates to decoding apparatus and method and, more particularly, to data processing apparatus and method which decode code encoded in image object units.

[0002] Further, the present invention relates to data processing apparatus and method which process a data array constructing an image with a plurality of coded image objects.

DESCRIPTION OF RELATED ART

[0003] In recent years, with advancement in image encoding techniques and progress of computer capabilities, an encoding method to separate an image into objects and encode by each object has been proposed. The image encoding in object units enables optimum encoding by each object, thus improving the coding efficiency. At the same time, a function to generate a new image by editing the objects within the image can be obtained. For example, in the technology of still image, a method to separate an image into "character", "line", "frame", "image", "table" and "background", and perform optimum encoding on the respective areas, such as the ACBIS method (by Maeda, and Yoshida in "The 1996 Institute of Electronics, Information and Communication Engineers General Conference D-292") has been proposed. According to this method, the JBIG (Joint Bi-level Image Group) encoding as a binary-image encoding method is performed on the "character", "line", "frame" and "table" areas, and in the "background" area, its representative value is encoded.

[0004] Further, in a moving image, a method to perform encoding in object units has been studied as an international standard method, MPEG4 (Moving Picture Experts Group phase 4) (Eto, "MPEG4 Standardization" (The Journal of The Institute of Image Electronics Engineers of Japan, vol. 25, No. 3, 1996, pp. 223-228). Fig. 1 shows an example of a frame of a moving image to be encoded by the MPEG4 coding. In Fig. 1, a frame 20 comprises four objects as shown in Fig. 2, i.e., a background object 28, an object 21 representing a helicopter, an object 22 representing a train, and an object 23 representing a car. To indicate the shapes of the objects except the background, each object is masked such that a black part of a rectangular area surrounding the object is an "outer area", and a white part is an "inner area" (24 to 26 in Fig. 2), and by this masking, an arbitrary shaped object can be handled.

[0005] Fig. 3 shows a construction for coding in object units. An input image 1 is inputted into an object segmenter 2, and is separated into respective objects. For example, the image in Fig. 1 is separated by the object segmenter 2 into the objects 28, 21, 22 and 23, and the objects are independently encoded. That is, an object encoder 3 encodes the object 28; an object encoder 4, the object 21; an object encoder 5, the object 22; and an object encoder 6, the object 23. A multiplexer 7 multiplexes code data outputted from the object encoders 3 to 6, and outputs the multiplexed data as code data 8.

[0006] Fig. 4 shows a construction for decoding an image encoded in object units. The code data 8 is inputted into a demultiplexer 9, and separated into code data corresponding to the respective objects. The separated code data are independently decoded. That is, an object decoder 10 decodes the object 28; an object decoder 11, the object 21; an object decoder 12, the object 22; and an object decoder 13, the object 23. An object compositer 14 arranges image data outputted from the object decoders 10 to 13 in proper positions for the respective objects, thus composes them as one image, and outputs the image data as a reproduced image 15.

[0007] In moving image coding represented by the MPEG2 (Moving Picture Experts Group phase 2) standard, coding is made in frame or field units. To realize reuse or editing of contents (person, building, voice, sound, background and the like) constructing a video image and audio data of a moving image, the MPEG4 standard is characterized by handling video data and audio data as objects. Further, objects included in a video image area independently encoded, and the objects are independently handled.

[0008] Fig. 25 shows an example of the structure of object code data. The moving image code data based on the MPEG4 standard has a hierarchical structure, from the point of improvement in coding efficiency and editing operability. As shown in Fig. 25, the head of code data has a visual_object_sequence_start_code (VOSSC in Fig. 25) for identification. Then, code data of respective visual objects follows, and visual_object_sequence_end_code (VOSEC in Fig. 25) indicative of the rear end of the code data is positioned at the end. As well as obtained moving images, computer graphics (CG) data and the like are defined as visual objects.

[0009] The visual object data has visual_object_start_code (Visual Object SC in Fig. 25) for identification at its header, then profile_and_level_indication (PLI in Fig. 25) indicative of an encoding level. Then, information on visual objects, is_visual-object_identifier (IVOI in Fig. 25), visual_object_varid (VOVID in Fig. 25), visual_object_priority (VOPRI in Fig. 25), visual_object_type (VOTYPE in Fig. 25) and the like follow. These data construct header information

of the visual object. "VOTYPE" has a value "0001" if the image is a moving image obtained by image pickup. Then, video object (VO) data as a cluster of moving image code data follows.

[0010] The VO data is code data indicative of each object. The VO data has video_object_start_code (VOSC in Fig. 25) for identification at its header, further, the VO data has video object layer data (VOL data in Fig. 25) to realize scalability. The VOL data has video_object_layer_start_code (VOLSC in Fig. 25) and video object plane data (VOP data in Fig. 25) corresponding to one frame of moving image. The VOL data has video_object_layer_width (VOL_width in Fig. 25) and video_object_layer_height (VOL_height in Fig. 25) indicative of size, at its header. Also, the VOP data has video_object_plane_width (VOP_width in Fig. 25) and video_object_plane_height (VOP_height in Fig. 25) indicative of size, at its header. Further, the header of the VOL data has bit_rate code indicative of bit rate.

[0011] Note that in each layer of the code data structure, data of an arbitrary length which starts with user_data_start_code can be inserted by a user. The user data is distinguished from the code data by recognition of start code VOSC and VOLSC or VOPSC following the user data.

[0012] Further, arrangement information, which is information to arrange the respective objects on the decoding side, is called a system code. In the system code, similar to VRML (Virtual Reality Markup Language) as a CG language, information describing arrangement of divided objects, reproduction timing or the like is encoded. The system code describes the relation among the respective objects with conception of nodes. Hereinbelow, the nodes will be specifically described with reference to Figs. 26 and 27.

[0013] Fig. 26 is an example of an image constructed with a plurality of objects. This image comprises a Background object 2000, a Balloon object 2001, a Bird object 2002, a Jet object 2003, a Car object 2004, a Woman object 2005 and a Man object 2006, respectively representing background, a balloon, a bird, an airplane, a car, a woman and a man.

[0014] Fig. 27 shows a node tree in the image in Fig. 26. The entire image is represented by a Scene node. The Scene node is connected to the Background object 2000, the Car object 2004, and a People node indicative of people and a Fly node indicative of things flying in the sky. Further, the People node is connected to the Woman object 2005 and the Man object 2006. The Fly node is connected to the Balloon object 2001, the Bird object 2002 and the Jet object 2003. The relation among the objects is described in the data of the system code.

[0015] In this manner, according to the MPEG4 standard, by handling objects in a moving image independently, the decoding side can freely arrange various objects. Further, in broadcasting companies, contents producing companies and the like, by generating code data of objects beforehand, a very large number of moving image data can be generated from limited contents.

[0016] However, the above-described techniques have the following problems. To decode respective objects independently, decoders corresponding to the number of separated objects are required. However, on the decoding side, it is impossible to prepare an arbitrary number of decoders. Accordingly, the number of independently encoded objects may be larger than the number of prepared decoders. The decoding apparatus as shown in Fig. 5 has three object decoders. A demultiplexer 9 allocates the object decoders to the code data 8 in input order. If the code data 8 includes four objects, the demultiplexer 9 allocates the object 28 to the object decoder 10, the object 21, to the object decoder 11, and the object 22, to the object decoder 12. However, regarding the object 23, as there is no available object decoder, the object 23 is not decoded. Accordingly, in an image obtained by decoding the objects and synthesizing them, the object 23 is omitted, as in a frame 38 in Fig. 6.

[0017] That is, in the coding based on the MPEG4 standard, as an unspecified number of objects are handled, the number of decoding means to decode all the objects cannot be determined especially on the decoding side, accordingly, it is very difficult to construct an apparatus or system. For this reason, in the standardized MPEG4 coding, to determine the specifications upon designing of code data and encoder/decoder, the concepts of profile and level are defined and the number of objects and the upper limit value of bit rate are provided as coding specifications. Fig. 28 shows an example of a profile table defining the number of objects and the bit rate upper limits of profiles and levels.

[0018] In the MPEG4 standard, a coding tool differs in accordance with profile. Further, as shown in Fig. 28, the amount of code data of handled image is determined stepwisely in accordance with level. Note that the maximum number of objects to be handled and the maximum bit rate value are upper limits in the coding specifications, and all the values are included in the coding specifications as long as they are less than the above maximum values. For example, in a case where a coding tool is available in a Core profile, the number of objects is six, and coding is performed at a bit rate of 300 kbps, the code data and the coding tool correspond to level 2 (Core profile and level 2).

[0019] The above-described profile and level are indicated in the PLI in a bit stream of MPEG4 code data as shown in Fig. 25. Accordingly, a decoder which decodes a bit stream of MPEG code data can determine whether or not decoding is possible by referring to the PLI. The decoding is impossible in the following case.

[0020] For example, a decoder corresponding to of Core profile and level 1 cannot decode code data of Core profile and level 2 since the maximum bit rate of Core profile and level 2 is 2000 kbps, far higher than 384 kbps as the maximum bit rate of Core profile and level 1.

[0021] Further, in an image including four objects, by synthesizing two code data of Simple profile and level 1, two code data of Simple profile and level 2 can be generated. However, as the maximum number of objects of Simple profile

and level 2 is 4, code data which cannot belong to any profile or level of the MPEG4 standard is generated. Accordingly, such coded data cannot be decoded.

[0022] Further, for example, if a new bit stream is generated by multiplexing two code data of Simple profile, with bit rates 48 kbps and 8 kbps, of two images respectively including two objects, the bit rate of the new bit stream may be over 64 kbps. In this case, the level of the code data must be raised to level 2, and it cannot be decoded by a decoder of Simple profile and level 1.

[0023] That is, if the coding specifications (profile and level) of a decoder do not sufficiently cover the coding specifications (profile and level) of code data, the decoder cannot decode the code data.

[0024] This problem becomes especially outstanding upon synthesizing a plurality of images. For example, when a plurality of code data, respectively decodable by a decoder are synthesized, occasionally the decoder cannot decode the synthesized code data. Further, if the synthesized code data does not correspond to any of MPEG4 Profiles and levels, it cannot be decoded by a decoder based on the MPEG4 standard.

SUMMARY OF THE INVENTION

[0025] The present invention has been made to solve the above-described problems, and has as a concern to provide data processing apparatus and method which decode all the image objects even if the number of decoders is limited.

[0026] According to the present invention there is provided a data processing apparatus having decoding means for decoding code encoded in image object units, said apparatus comprising: detection means for detecting the number of objects included in input code and the number of objects decodable by said decoding means; and control means for controlling the number of objects of the input code, based on the number of objects and the number of decodable objects detected by said detection means.

[0027] Further, another concern of the present invention is to provide data processing apparatus and method which decode coded still image and/or moving image without degrading the image quality even if the number of decoders is limited.

[0028] According to a feature of the present invention, the above-described apparatus further comprises; extraction means for extracting location information of the objects included in said code; and combining means for combining code of a plurality of objects, based on an instruction from said control means and the location information extracted by said extraction means.

[0029] Further the above-described apparatus may comprise: extraction means for extracting motion information indicative of motions of the objects included in said code; and combining means for combining a plurality of objects based on an instruction from said control means and the motion information extracted by said extraction means.

[0030] Further, another concern of the present invention is to provide data processing apparatus and method which decode code data, encoded by each of plural image objects, with decoders of arbitrary coding specifications.

[0031] Further, another concern of the present invention is to provide data processing apparatus and method which control the number of objects included in code data.

[0032] According to an aspect of the present invention, there is provided a data processing apparatus for processing a data array to reproduce an image with a plurality of coded image objects, said apparatus comprising: detection means for detecting the number of image objects included in said data array; and control means for controlling the number of image objects included in said data array based on the number of image objects detected by said detection means.

[0033] Further, another concern of the present invention is to provide data processing apparatus and method which synthesize a plurality of code data, encoded by each of plural image objects, to obtain one code data based on a predetermined coding standard.

[0034] According to another aspect of the present invention there is provided a data processing apparatus comprising: input means for inputting a plurality of image data to construct one frame, wherein said image data respectively including N image objects, where $N \geq 1$ holds; and generation means for generating image data having M image objects, where $M \geq 1$ holds, constructing said one frame, by integrating at least a part of said N image objects based on additional information indicative of relation among the image objects.

[0035] Further, another concern of the present invention is to provide data processing apparatus and method which decode synthesized code data with decoders of arbitrary coding specifications.

[0036] Further, another concern of the present invention is to provide data processing apparatus and method which control the number of objects included in code data and/or the information amount of the code data.

[0037] According to another aspect of the present invention, there is provided a data processing apparatus for processing a data array to reproduce one frame image with a plurality of coded image objects, said apparatus comprising: input means for inputting a plurality of data arrays; instruction means for instructing synthesizing of a plurality of data arrays inputted by said input means; designation means for designating coding specifications of a processed data array; control means for controlling information amounts of the plurality of data arrays inputted by said input means, based on the coding specifications designated by said designation means; and synthesizing means for synthesizing the

plurality of data arrays with information amounts controlled by said control means, based on the coding specifications designated by said designation means..

[0038] Other features and advantages of the present invention will be apparent from the following description taken in conjunction with the accompanying drawings, in which like reference characters designate the same name or similar parts throughout the figures thereof.

BRIEF DESCRIPTION OF THE DRAWINGS

[0039] The accompanying drawings, which are incorporated in and constitute a part of the specifications, illustrate embodiments of the invention and, together with the description, serve to explain the principles of the invention.

Fig. 1 is an example of the image processed by the MPEG4 coding;
Fig. 2 is an explanatory view showing the objects of the image in Fig. 1;
Fig. 3 is a block diagram showing the construction for coding in object units;
Fig. 4 is a block diagram showing the construction for decoding an image encoded in object units;
Fig. 5 is a block diagram showing the construction for decoding an image encoded in object units;
Fig. 6 is an example of decoded image where an object is omitted;
Fig. 7 is a block diagram showing the construction of a decoding apparatus according to the present invention;
Fig. 8 is an example of 1-frame code data;
Fig. 9 is an example of synthesized code data;
Fig. 10 is a block diagram showing the construction of an object combiner according to a first embodiment of the present invention;
Figs. 11A and 11B are examples of object combining;
Fig. 12 is an example of 1-frame code data to be motion-compensated;
Fig. 13 is an example of synthesized code data;
Figs. 14A to 14C are examples of objects of a still image and combined objects;
Fig. 15 is an example of 1 frame of a moving image;
Fig. 16 is a block diagram showing the construction of the object combiner according to a second embodiment of the present invention;
Figs. 17A and 17B are examples of combined objects;
Fig. 18 is an example of code data including combined objects;
Fig. 19 is a block diagram showing the construction of the object combiner according to a third embodiment of the present invention;
Fig. 20 is an example of input code data;
Fig. 21 is an example of processed code data;
Fig. 22 is a block diagram showing the construction of the object combiner according to a fourth embodiment of the present invention;
Fig. 23 is a block diagram showing the construction of the object combiner according to a modification;
Fig. 24 is a block diagram showing the construction of the object combiner according to another modification;
Fig. 25 is an example of the structure of object code data;
Fig. 26 is an example of the image constructed with a plurality of objects;
Fig. 27 is an example of a node tree in the image in Fig. 26;
Fig. 28 is an example of the profile table defining the number of objects and the bit rate upper limits by profile and level;
Fig. 29 is a block diagram showing the construction of a moving image processing apparatus according to a fifth embodiment of the present invention;
Fig. 30 is a block diagram showing the construction of a profile and level regulator according to the fifth embodiment;
Figs. 31A and 31B are examples of the structure of code data of moving image;
Fig. 32 is a block diagram showing the construction of the profile and level regulator according to a sixth embodiment of the present invention;
Fig. 33 is a block diagram showing the construction of the profile and level regulator according to a seventh embodiment of the present invention;
Fig. 34 is a block diagram showing the construction of an object integrator according to the seventh embodiment;
Fig. 35 is an example of the structure of synthesized code data according to the seventh embodiment;
Fig. 36 is a block diagram showing the construction of the object integrator according to a modification of the seventh embodiment;
Fig. 37 is an example of synthesized color image information according to the seventh embodiment;

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0040] Preferred embodiments of the present invention will now be described in detail in accordance with the accompanying drawings.

First Embodiment

[Construction]

[0041] Fig. 7 is a block diagram showing the construction of a decoding apparatus according to the present invention. Note that elements approximately corresponding to those in Figs. 3 and 5 have the same reference numerals and detailed explanations of the elements will be omitted.

[0042] In Fig. 7, an object number regulator 40 includes an object counter 41 which counts the number s of objects, an object decoder counter 42 which counts the number d of object decoders, and an object combiner 43 which combines the plurality of objects included in the code data 8. Numeral 50 denotes a storage device comprising a magnetic disk or the like.

[0043] The code data 8 inputted into the decoding apparatus is subjected to coding by an encoder as shown in Fig. 3, for example. The code data 8 includes four objects at the maximum. Hereinbelow, description will be made using a moving image frame as shown in Fig 1 as an original image.

[0044] The code data 8 is inputted into the object number regulator 40 by each frame. When code data of a frame has been inputted, the object counter 41 counts the numbers of objects included in the code data.

[0045] Fig. 8 is an example of 1-frame code data. The code data has "Header" indicative of the attribute of the frame at its head, next, code data indicative of background object (Object 0 in Fig. 8). Then, code data of the respective objects, i.e., code data of the object 21 (object 1), the object 22 (Object 2) and the Object 23 (Object 3) follow. The code data of each object comprises Start code (SC) indicative of the header of the object, Location (Loc) code indicative of location of the object, Size code indicative of the size of the object, Shape code indicative of the shape of the object, and Texture code indicative of the object itself.

[0046] In the following description, the Shape code is binary-encoded by MR coding, and the Texture code is encoded by block-coding. Note that block encoding is dividing an object into, e.g., 8×8 pixel blocks, then performing discrete cosine transformation (DCT) on each block, and quantizing and encoding the obtained conversion coefficients (DCT coefficients), such as JPEG coding. Fig. 8 shows the Texture code of the Object 1. The Texture code is a set of block-based code, DCT-COEFs. The code DCT-COEF is obtained by one-dimensionally rearranging quantization values of DCT coefficients and encoding quantization values other than zero run-length and zero value. If all the quantization val-

ues are zero, no DCT-COEF is generated.

[0047] The count value s of the object counter 41 is reset to zero upon start of input of 1-frame code data. Then, the number of occurrence of SC indicative of the header of object of the code data is counted. The result of counting is inputted into the object combiner 43. The object decoder counter 42 counts the number d of object decoders. In the present embodiment, as the number of object decoders is three, accordingly, the output from the object decoder counter 42 is "3".

[0048] Fig. 10 is a block diagram showing the construction of the object combiner 43. A terminal 61 inputs the code data 8. A terminal 62 inputs the number s of the objects from the object counter 41. A terminal 63 inputs the number d of the object decoders from the object decoder counter 42.

[0049] A code memory 64 is used for storing code data for one or more frames inputted from the terminal 61. A location information extractor 65 extracts the Loc code from the code data stored in the code memory 64, and stores the extracted Loc code into a location information memory 66 in frame units. An object distance calculator 67 calculates distances between respective objects, based on the Loc code stored in the position location memory 66.

[0050] A distance comparator 68 compares the distances calculated by the object distance calculator 67, and selects objects to be combined, based on the result of comparison. A selector 69 outputs object code data, read from the code memory 64, to a code divider 70 or terminal 76, designated by the distance comparator 68 as an output destination, for each object.

[0051] The code divider 70 divides object code data into Loc, Size, Shape and Texture code. A location code combiner 71 combines the Loc code of two objects into one Loc code. A size code combiner 72 combines the Size code of two objects into one Size code. A shape code combiner 73 combines the Shape code of two objects into one Shape code. A texture code combiner 74 combines the Texture code of two objects into one Texture code. A code synthesizer 75 synthesizes outputs from the location code combiner 71 to the texture code combiner 74 into one code data.

[0052] One of the output from the code synthesizer 75 and that from the selector 69 is forwarded to the next stage via a terminal 76.

[Operation]

[0053] Next, the operation of the present embodiment will be described on a case where an MPEG Intra frame or respective Motion JPEG frames are independently encoded.

• Frame-Based Coding

[0054] In Fig. 10, code data for one or more frames is stored via the terminal 61 into the code memory 64, and the number s of objects is inputted via the terminal 62 from the object counter 41. In case of the code data in Fig. 8, the number s of objects is four (s=4). Further, the number d of object decoders is inputted via the terminal 63 from the object decoder counter 42. In the present embodiment, the number d of object decoders is three (d=3). Accordingly, s-d=1 holds, i.e., the construction has one less decoder.

[0055] The object distance calculator 67 obtains the distance between the object 21 and the object 22 from the Loc code stored in the location information memory 66. If the location of the object 21 is (x1,y1) and that of the object 22 is (x2,y2), the distance D12 between these objects is represented by the following equation:

$$D12 = \sqrt{\{(x1-x2)^2 + (y1-y2)^2\}} \tag{1}$$

[0056] Similarly, the distance D13 between the abject 21 and the object 23 and the distance D23 between the object 22 and the object 23 are obtained. Based on the obtained distances between objects, the distance comparator 68 selects a plurality of objects with a short distance therebetween, to combine the objects for compensation of the shortage of object decoder. For example, the distance comparator 68 selects a plurality of objects with the smallest sum of distance therebetween in a plurality of frames. In the present embodiment, as the shortage of object decoder is "1", two objects are combined into one object. If the sum of distance D12 is the smallest, the object 21 and the object 22 are combined. Thus, the shortage of object decoder is resolved.

[0057] If the object 21 and the object 22 are combined, the selector 69, controlled by the output from the distance comparator 68, sends the header outputted from the code memory 64 to the terminal 76, and sends the Object 0 as background object to the terminal 76.

[0058] Next, the output from the distance comparator 68 for the Object 1 corresponding to the object 21 indicates "selection", accordingly, the selector 69 sends the Object 1 to the code divider 70. The Loc code of the Object 1 is sent to the location code combiner 71, the Loc code and Size code are sent to the size code combiner 72, the shape code is sent to the shape code combiner 73, and the Texture code is sent to the texture code combiner 74. Next, the output from the distance comparator 68 for the Object 2 corresponding to the object 22 also indicates "selection", accordingly,

the Object 2 code is divided into the respective code, and the divided code are inputted into the location code combiner 71 to the texture code combiner 74, as in the case of the Object 1.

[0059] Note that as the output from the distance comparator 68 for the Object 3 corresponding to the object 23 indicates "non-selection", the code data of the object 23 is outputted to the terminal 76 without any processing. Further, for a frame having the code data 8 where $s\text{-}d>0$ holds, object combining is not performed, and the output from the selector 69 is forwarded to the terminal 76.

[0060] The location code combiner 71 decodes the respective Loc code, and obtains location information (x1,y1) to (xn,yn) of the plurality of objects. Then, as represented by the following equation, the location code combiner 71 selects the minimum value of x- and y-coordinates from these location information, and outputs new location information (x1' ,y1').

$$(x1',y1') = (\min(x1,x2,...,xn), \min(y1,y2,...,yn)) \qquad (2)$$

n: the number of combined objects

[0061] The size code combiner 72 decodes the respective Loc and Size code, and obtains location information and size information (x1,y1), (Sx1,Sy1) to (xn,yn), (Sxn,Syn) of the plurality of objects. Then, the size code combiner 72 calculates new location information (x1',y1') from the equation (2), and obtains new size information (Sx1',Sy1') from the following equation and outputs the information.

$$(Sx1',Sy1') = (\max(x1+Sx1,x2+Sx2,...,xn+Sxn)-x1', \max(y1+Sy1,y2+Sy2,...,yn+Syn)-y1') \qquad (3)$$

[0062] The shape code combiner 73 generates code synthesized from the shapes of the plurality of objects. When the objects 21 and 22 are synthesized, the shape of a new object is represented by a mask 80 as shown in Fig. 11A. The original masks 24 and 25 remain the same, and the portion other than the masks are newly added. In Fig. 11A, the value of the hatched portion is the same as that of the solid black masks 24 and 25. Accordingly, as zero-run has increased on the right side of the mask 24, zero run-length is added after code indicative of a change point nearest to the frame right end.

[0063] Further, if another object does not exist on the right side of the object 21, the above-described change point merely indicates the final change point of the line, and the code does not increase. On the other hand, if another object exists on the right side of the object 21, zero run-length corresponding to the number of pixels between both objects is added to the code. That is, the code can be replaced with code to which zero run-length is added. Further, if there is a third object on the right side of the other object on the right side of the object 21, the code of the object 21 is replaced with code where the zero run-length corresponding to the interval between the objects have been added the code of the object 21. The replaced code is outputted as new Shape code. Note that with respect to a line including no object, almost no code is generated.

[0064] The texture code combiner 74 generates code synthesized from textures of the plurality of objects. Fig. 11B shows a status where the texture of the object 21 and that of the object 22 are synthesized. The texture of a new object is represented as an object 81. The original objects 21 and 22 remain the same, and a hatched portion other than the objects is newly added. Note that the value of the hatched portion is zero. In the MPEG coding or the like, the DC component of a pixel of interest is converted into a difference between the DC component and that of a left block. Further, quantization values of AC components are one-dimensionally arrayed, and zero run-length and nonzero values are encoded. In the hatched portion in Fig. 11B, the difference between the DC component of a pixel of interest and that of a left block is zero, and the values of all the AC components are zero. In this case, in the MPEG1 coding, in macro-block units, 1 bit indicative of macro block type, luminance 12 bits and chromaticity 4 bits indicative of DC component size, and EOB (End of Block) 12 bits indicative of the end of the block, i.e., total 29 bits, are added. In this manner, Texture code of the object 81 where the textures of the plurality of objects are combined is generated, and the Texture code is outputted.

[0065] The code synthesizer 75 synthesizes outputs from the location code combiner 71 to the texture code combiner 74, to generate code data of the combined object, and outputs the code data to the terminal 76.

[0066] Fig. 9 shows an example of code data synthesized as above. The code data Object 1 of the object 21 and the code data Object 2 of the object 22 in Fig. 8 are combined into code data Object 1'. Note that the code data Object 3 of the object 23 remains the same.

[0067] The code data processed as above is inputted into the demultiplexer 9, and the object code data is divided into Object 0, Object 1' and Object 3. Then, the code data Object 0 is inputted into the object decoder 10; the code data Object 1' is inputted into the object decoder 11; and the code data Object 3 is inputted into the object decoder 12. The respective object decoders output location information obtained by decoding the code data and the image data, to the object compositer 14. The object compositer 14 arranges the image data in accordance with the location information of

the respective objects, to obtain a reproduced image 15.

•Moving Image Coding

[0068] In moving image coding, the coding efficiency is improved by motion compensation. As an example, coding by using the correlation between frames such as a predicted frame in MPEG standard will be described.

[0069] Fig. 12 is an-example of code data 8 for 1 frame to be motion-compensated. Similar to the code data in Fig. 8, the code data in Fig. 12 has a header, code data Object 0 indicative of background object, and code data (Object 1 to Object 3) of respective objects. Each code data comprises the SC indicative of the header of the object, the Loc code indicative of location, the Size code indicative of size, the Shape code indicative of shape and Texture code indicative of texture. In the MEGP coding, an object is divided into macro blocks and motion compensation is performed in block units. As a result, The Texture code comprises MV code indicative of motion vector as a result of motion vector coding, and DCT-COEF quantized and encoded from the result of block-based division and DCT conversion.

[0070] In moving image coding, the object number regulator 40 performs similar processing to that in frame-based coding as described above. Although detailed explanation will be omitted, if the shortage of object decoder is "1" and the distance D12 is the shortest, the objects 21 and 22 are combined, to reduce the number of objects. Further, the motion vectors included in the respective objects and the result of coding of DCT coefficients are not changed.

[0071] In the MPEG coding or the like, when frame correlation is utilized, the DC component and AC components of predicted difference are encoded. Further, a motion vector is converted to the difference between the motion vector and that of a left macro block. Accordingly, in the hatched portion in Fig. 11B, the difference of the motion vector is zero, and the values of all the AC components are zero. In the MPEG coding, such macro block is not encoded and is skipped, and corresponding code merely indicates the number of skipped macro blocks. Accordingly, in the object combining, the code of macro block included in an object which appears next is merely changed, and the code is merely slightly changed. In this manner, the Texture code of combined objects is generated and outputted.

[0072] Fig. 13 is an example of code data synthesized as above. The code data Object 1 of the object 21 and the code data Object 2 of the object 22 in Fig. 12 are combined into code data Object 1'. Note that the code data Object 3 of the object 23 remains the same.

[0073] As described above, in the present embodiment, in a moving image encoded in object units, if the number of coded objects is greater than that of decoders, i.e., there is shortage of decoders, to reduce the number of objects in correspondence with the shortage, a plurality of objects with a short distance therebetween are combined. This enables efficient and proper reproduction of moving image including a number of coded objects, with a limited number of decoders. Further, the objects are synthesized in code data status. That is, as the combining is made by change or addition of code, the objects can be synthesized at a high speed, and further, increment in code length is very small.

[0074] Further, in the above description, a coded moving image is decoded, however, a coded still image can be similarly processed. That is, the above-described frame-based decoding can be applied to still image decoding. For example, as shown in Fig. 14A, an image 90 includes character areas 91 and 94 and photographic areas 92 and 93, and the characters are encoded by the MMR (Modified Modified Read) coding and the photographs are encoded by the JPEG coding. If only one decoder for the MMR coding and only one decoder for the JPEG coding are prepared, the image 90 can be decoded by combining and dividing the respective areas into objects 95 and 96 as shown in Figs. 14B and 14C.

[0075] Note that in the above description, the Shape code is encoded by the MMR coding, and the Texture code is encoded by the MPEG coding, however, the present invention is not limited to these coding methods. Further, the function of the demultiplexer 9 may be incorporated into the object number regulator 40. Further, if $s\text{-}d \geq 2$ holds as the difference between the number d of object decoders and the number s of objects included in code data, objects with a short distance therebetween, i.e., $2 \cdot (s\text{-}d)$ objects may be combined with (s-d) objects, or (s-d+1) objects with the shortest distance therebetween may be combined into one object.

[0076] In the above description, code data of regulated number of objects are inputted into the object decoders. However, if it is arranged such that the code data of regulated number of objects are temporarily stored in the storage device 50 as shown in Fig. 7, and the code data of regulated number of objects are read from the storage device 50 and decoded, decoding processing can be performed at a speed higher than that in decoding with object combining.

[0077] Further, according to the present embodiment, the number of object decoders is not limited. Accordingly, the processing capability can be easily improved by increasing the object decoders. Further, the location of an object may be obtained precisely by decoding, e.g., the Shape code, as well as utilizing the location code.

Second Embodiment

[Construction]

[0078] Fig. 16 is a block diagram showing the construction of the object combiner 43 according to a second embodi-

ment of the present invention. In Fig. 16, elements corresponding to those in the construction of Fig. 10 have the same reference numerals, and detailed explanations of the elements will be omitted.

[0079] Numerals 111 and 112 denote code memories having a similar function to that of the code memory 64; 113 and 114, location information memories having a similar function to that of the location information memory 66; and 115, an object motion calculator which detects the movement of object.

[0080] Numeral 116 denotes a determination unit 116 which determines whether or not object combining is necessary and determines objects to be combined, based on the results of calculation by the object motion calculator 115 and the object distance calculator 67, and the number s of objects and the number d of object decoders inputted from the terminals 62 and 63. Numeral 117 denotes a selector which outputs object code data read from the code memory 111, or from the code memory 112, if necessary, to an output destination designated by the determination unit 116.

[Operation]

[0081] Next, the operation of the object combiner 43 as shown in Fig. 16 will be described. First, frame-based coding to independently encode respective frames such as Intraframe coding in the MPEG standard or Motion JPEG coding will be described.

[0082] In this case, the code memory 111 has the same function as that of the code memory 64 in Fig. 10; the location information memory 113 has the same function as that of the location information memory 66 in Fig. 10; and the determination unit 116 has the same function as that of the distance comparator 68 in Fig. 10. Accordingly, the code data for 1 frame inputted from the terminal 61 is stored into the code memory 111. The number s of objects, as the output from the object counter 41, is inputted into the terminal 62. The number d of object decoders, as the output from the object decoder counter 42, is inputted into the terminal 63. The location information extractor 65 extracts location information of respective objects from the code data stored in the code memory 111, and inputs the extracted information into the location information memory 113. The object distance calculator 67 calculates distances between objects based on the location information stored in the location information memory 113. The determination unit 116 determines whether or not object combining is necessary from the number s of objects and from the number d of object decoders. If object combining is necessary, the determination unit 116 determines the number of objects to be combined, then compares the distances between objects obtained by the object distance calculator 67, and determines the necessary number of objects to be combined.

[0083] The object code data read from the code memory 111 is inputted into the selector 117. The selector 117 forwards code data of header, background object and uncombined objects to the terminal 76. On the other hand, the selector 117 inputs code data of objects to be combined to the code divider 70. The code data inputted into the code divider 70 is divided into location, size, shape and texture code data, and inputted into the location code combiner 71, the size code combiner 72, the shape code combiner 73 and the texture code combiner 74, respectively. Object code data, combined in a procedure similar to that described in the first embodiment, is outputted from the terminal 76.

[0084] Next, a frame encoded by using the correlation between frames such as a predicted frame in the MPEG coding, will be described. In this case, the MPEG-coded frame 20 in Fig. 1 and a frame 100 shown in Fig. 15, following the frame 20, will be described. Note that in the frame 100, the object 21 (helicopter) has moved rightward, and the object 22 (train) and the object 23 (car) have moved leftward, with respect to the frame 20.

[0085] Prior to processing, the code memories 111 and 112 and the location information memories 113 and 114 are cleared, and the other elements are initialized. The number s of objects is inputted into the terminal 62, and the number d of object decoders is inputted into the terminal 63. First, the code data of the frame 20 is inputted into the terminal 61, and stored into the code memory 111. The location information extractor 65 stores location information of the respective objects in the frame 20 into the location information memory 113. The object distance calculator 67 obtains distances between the respective objects in the frame 20.

[0086] Next, the code data in the code memory 111 is moved to the code memory 112, and the location information in the location information memory 113 is also moved to the location information memory 114, then the code data of the frame 100 is inputted into the terminal 61 and stored into the code memory 111. The location information extractor 65 stores location information of the respective objects in the frame 100 into the location information memory 113.

[0087] The object motion calculator 115 calculates the motions of the respective objects from the locations of the respective objects in the location information memories 113 and 114. In case of the object 21, assuming that its location in the frame 20 is (x211,y211) and that in the frame 100 is (x212,y212), the motion vector MV21 = (mv21x,mv21y) is represented as:

$$MV21 = (mv21x,mv21y) = ((x212-x211),(y212-y211)) \tag{4}$$

[0088] Regarding the objects 22 and 23, motion vectors MV22 and MV23 are obtained in a similar manner.

[0089] The distances D12, D13 and D23 as the outputs from the object distance calculator 67, and the motion vectors

MV21, MV22 and MV23 as outputs from the object motion calculator 115 are inputted into the determination unit 116. The determination unit 116 determines whether or not object combining is necessary, from the number s of objects and the number d of object decoders, and if the object combining is necessary, determines the number of objects to be combined and objects to be combined.

[0090] In this case, objects having motion vectors with directions close to each other are determined as objets to be combined. Then, the difference vectors between the motion vectors of the respective objects are obtained, and motion vector(s) less than a threshold value Thdv is selected. That is, the difference vector DV2122 between the motion vector MV21 of the object 21 and the motion vector MV22 of the object 22 is represented by the following equation:

$$DV2122 = (dv2122x,dv2122y) = ((mv21x-mv22x),(mv21y-mv22y)) \quad (5)$$

[0091] The size D2122 of the difference vector DV2122 is represented by the following equation:

$$D2122 = \sqrt{(dv2122x^2 + dv2122y^2)} \quad (6)$$

[0092] All the difference vector sizes are obtained. The obtained difference vector sizes D2122, D2223 and D2123 are compared with the threshold value Thdv, and the difference vector size(s) less than the threshold value is selected. As the objects 22 and 23 have moved in the same direction, the difference vector size D2223 of the difference vector is less than that with respect to the object 21. If only the difference vector size D2223 is less than the threshold value Thdv, the objects to be combined are the objects 22 and 23. If all the difference vector sizes are less than the threshold value Thdv, objects with the shortest distance therebetween are selected as objects to be combined. Further, if there is no difference vector size less than the threshold value, objects with the shortest difference therebetween are combined.

[0093] Then, object combining is performed based on the determination. In this case, the object 22 and the object 23 are combined so as to reduce the number of objects. This operation will be described using the code data 8 in Fig. 8 as an example.

[0094] First, the selector 117 reads the header from the code memory 112 and outputs the header via the terminal 76. Further, the selector 117 reads the code data Object 0 of the background object, and outputs the code data via the terminal 76. As the object 21 is not combined, the selector 117 similarly reads the code data Object 1 and outputs the code data via the terminal 76.

[0095] Then, as the next code data Object 2 corresponds to the object 22, the selector 117 reads the code data Object 2 from the code memory 112 and inputs the code data into the code divider 70. The code divider 70 inputs the Loc code from the object code data into the location code combiner 71, the Loc code and the Size code from the object code data, into the size code combiner 72, the Shape code from the object code data, into the shape code combiner 73, and the Texture code from the object code data, into the texture code combiner 74.

[0096] Next, the selector 117 reads the code data Object 3 of the object 23 to be combined with the object 22, from the code memory 112, and inputs the code data into the code divider 70. As in the case of the code data Object 2, divided code are respectively outputted.

[0097] The location code combiner 71 decodes the respective Loc code, and generates new location information (x2',y2') from the location information (x2,y2) and (x3,y3) of the two objects, based on the equation (2), then encodes the new location information (x2',y2'), and outputs the coded location information.

[0098] The size code combiner 72 decodes the Loc code and the Size code, then generates new size information (Sx2',Sy2') from the location information (x2,y2) and (x3,y3) and size information (Sx2,Sy2) and (Sx3,Sy3) of the two objects, based on the equation (3), then encodes the new size information (Sx2',Sy2'), and outputs the coded size information.

[0099] The shape code combiner 73 generates code of a shape synthesized from the shapes of the two objects. When the objects 22 and 23 are synthesized, the shape of the new object is represented by a mask 150 in Fig. 17A. That is, in Fig. 17A, a hatched portion is newly added to a mask 25 of the object 22 and a mask 26 of the object 23, as the mask 150. Note that the value of the hatched portion is the same as that of the mask 80 in Fig. 11A. Then, as in the case of the first embodiment, addition of zero-run and/or code change is performed, and the obtained code is outputted as new Shape code.

[0100] The texture code combiner 74 generates code of texture synthesized from the textures of the two objects. Fig. 17B shows a status where the texture of the object 22 and that of the object 23 are synthesized. That is, a texture having zero value, as represented as a hatched portion, is added to the textures of the objects 22 and 23. Then, as in the case of the first embodiment, code is added in macro block units in the hatched portion or the number of skipped blocks is changed, thus Texture code of an object 151 is generated and outputted.

[0101] The code synthesizer 75 synthesizes the outputs from the location code combiner 71 to the texture code combiner 74, to generate code data of the combined object. The code data of the combined object is outputted from the terminal 76.

[0102] Fig. 18 is an example of code data including the combined object. In Fig. 18, the code data Object 1 of the object 21 remains the same, while the code data Object 2 of the object 22 and the code data Object 3 of the object 23 are combined as code data Object 2'.

[0103] The generated code data is inputted into the demultiplexer 9, and divided into the code data Object 0, the Object 1, and the Object 2'. The code data object 0 is inputted into the object decoder 10; the code data Object 1 is inputted into the object decoder 11; and the code data Object 2' is inputted into the object decoder 12. The object decoders 10 to 12 decode the code data, generate location information and image data of the respective objects, and output them to the object compositer 14. The object compositer 14 arranges the image data in accordance with the location information of the respective objects, thus obtains the reproduced image 15.

[0104] In the present embodiment, in a moving image encoded in object units, if the number of coded objects is greater than that of decoders, objects are combined, from objects with motion vectors or moving amounts close to each other, whereby original image reproduction can be efficiently made even by a limited number of decoders. Further, as change or addition of code is performed in the form of code data, the processing can be made at a high speed, and increment in code length is very small. Further, as the decoding load on the respective decoders can be uniformed. Further, even if the currently-processed frame is not a frame encoded by the Intraframe coding, upon occurrence of scene change, objects to be combined in interframe coding can be re-determined.

[0105] In the present embodiment, the difference vector size and the distance between objects are used for determination of objects to be combined, however, the determination may be made by only using the difference vector size. Further, in the present embodiment, the Shape code is encoded by the MMR coding, and the Texture code is encoded by the MPEG coding, however, the present invention is not limited to these coding methods.

[0106] Further, in the present embodiment, the function of the demultiplexer 9 may be incorporated into the object combiner 40. Further, the number of object decoders and the number of objects included in code data are not limited to those in the embodiment. As long as $(s\text{-}d) \geq 2$ holds, $2 \cdot (s\text{-}d)$ objects can be combined to (s-d) objects, from objects with the minimum difference vector size, or (s-d+1) objects can be combined into one object from objects with the minimum difference vector size, or combination between the former and latter cases may be employed.

[0107] In the present embodiment, the decoding apparatus having decoders to outputs decoded results has been described, however, if it is arranged such that code outputted from the object combiner 43 is temporarily stored into the storage device 50, and the code read out of the storage device 50 is decoded, object combining is unnecessary, and high-speed decoding (image reproduction) is possible.

[0108] Further, in the present embodiment, as the number of object decoders can be freely set, the number of object decoders can be easily increased so as to improve processing capability. Further, the motion calculation may be made by referring to the motion vectors of objects as well as referring to the location information of the objects.

Third Embodiment

[0109] Fig. 19 is a block diagram showing the construction of the object combiner 43 according to a third embodiment of the present invention. In Fig. 19, elements corresponding to those in Fig. 10 have the same reference numerals, and detailed explanations of the elements will be omitted.

[0110] A code length extractor 200 extracts code lengths of respective objects of code data stored in the code memory 64, and stores the extracted code lengths into a code length memory 201. A code length comparator 202 compares the respective code lengths of the objects, stored in the code length memory 201, with each other, then determines whether or not object combining is necessary, and determines objects to be combined.

[0111] If object combining is performed, objects to be combined are determined, sequentially from objects with short code lengths. For example, if the number s of objects is four (s=4), and the number d of object decoder is three (d=3), s-d=1 holds, accordingly, two objects having short code lengths are combined into one object. If the code data of the frame 20 in Fig. 1 is as shown in Fig. 20, the code data Object 2 of the object 22 is the minimum code data, and the code data Object 1 of the object 21 is the next minimum code data. In this case, the code data Object 1 and Object 2 are combined. The operations of other elements are the same as those in the above embodiments. From the terminal 76, code data as shown in Fig. 21 is outputted.

[0112] Then, the code data Object 1 and Object 2 as the code data of the object 21 and the object 22 are combined in all the frames. The details of the combining are as described in the above respective embodiments. Even a motion-compensation frame or a frame encoded by the Intraframe-coding of the MPEG coding or the like are included in the image data, the object combining is made in a similar manner to that in the above respective embodiments.

[0113] According to the present embodiment, similar advantages to those in the above respective embodiments can be obtained. Further, in case of still image as shown in Fig. 14A, a character image is encoded at a high compression rate by the MMR coding, and the resulting code length is short. Accordingly, if an image where character portions are combined as shown in Fig. 14B is handled as one object, similar advantages to those as above can be obtained in the still image in Fig. 14A.

Fourth Embodiment

[0114] In the MPEG coding or the like, a frame within which coding is performed (a frame encoded by the Intraframe-coding) and a frame encoded by using interframe correlation (a frame encoded by the Interframe-coding) are treated. The frame encoded by the Intraframe-coding is inserted to ensure synchronism or to prevent accumulation of DCT differences.

[0115] The present embodiment re-determines objects to be combined upon coding of Intraframe-coding frame.

[0116] Fig. 22 is a block diagram showing the construction of the object combiner 43 according to a fourth embodiment of the present invention. In Fig. 22, elements corresponding to those in Fig. 10 have the same reference numerals, and detailed explanations of the elements will be omitted.

[0117] Numeral 301 denotes a header analyzer which analyzes the header of each frame. Numeral 302 denotes a distance comparator having approximately the same operation as that of the distance comparator 68 in Fig. 10. As in the case of the first embodiment, prior to processing, the number s of objects and the number d of object decoders are inputted. If s≤d holds, code inputted into the terminal 61 is outputted from the terminal 76 without any processing.

[0118] On the other hand, if s>d holds, the header of code data for 1 frame, inputted from the terminal 61 and stored into the code memory 64, is inputted into the header analyzer 301. In the header with description of frame attribute, information indicating whether or not the frame is a frame encoded by the Interframe-coding, i.e., a frame to be-encoded by using interframe correlation, is described. For example, the MPEG coding handles an I frame which is encoded within the frame without interframe correlation (by Intra coding), and a P and B frames encoded by using interframe correlation with motion-compensation.

[0119] When a frame encoded without interframe correlation is detected from the result of header analysis, the operation of the present embodiment is as follows. Code data is read out of the code memory 64. The location information extractor 65 extracts the Loc code following the SC of respective objects, and stores the extracted Loc code into the location information memory 66. The object distance calculator 67 obtains distances between the objects, and the distance comparator 302 selects, sequentially from objects with the shortest distance therebetween. Note that the procedure of selection is similar to that of the first embodiment. Information indicative of the selected objects are held in the distance comparator 302.

[0120] The information indicative of the selected objects held in the distance comparator 302 is updated only if a new instruction is inputted from the header analyzer 301, i.e., only if a frame encoded without interframe correlation has been newly detected.

[0121] On the other hand, in a frame encoded by using interframe correlation, object combining is performed in accordance with information indicative of selected objects held in the distance comparator 302, and code of a new object obtained by combining objects is outputted from the terminal 76, as in the case of the first embodiment.

[0122] In this manner, according to the present embodiment, objects to be combined are re-determined upon decoding of frame encoded by the Intraframe-coding, whereby change of coding efficiency by object combining can be suppressed. Even if a frame encoded by the Intraframe-coding is not detected, when scene change occurs, objects to be combined are re-determined, even with objects encoded by using interframe correlation. Regarding scene change, in a P frame, for example, if the number of macro blocks to be Intra-encoded is large, or in a B frame, if a frame where motion vectors are referred to greatly depends on its pervious or subsequent frame, it is determined that scene change has occurred.

[0123] According to the fourth embodiment, as in the case of the first embodiment, by re-determining objects to be combined in a frame encoded by the Intraframe-coding, change of coding efficiency due to object combining can be suppressed.

Modifications of First to Fourth Embodiments

[0124] As shown in Fig. 23, the header analyzer as described in the fourth embodiment can be added, as a header analyzer 401, to the construction in Fig. 16 of the second embodiment. That is, as a result of frame header analysis, if it is determined that the frame has been encoded without interframe correlation, a determination unit 402 determines objects to be combined, based on distances between objects outputted from the object distance calculator 67 and motions of objects outputted from the object motion calculator 115, as in the case of the second embodiment. Information indicative of objects to be combined is held in the determination unit 402, and only if an instruction is inputted from the header analyzer 401, the held content is updated.

[0125] As shown in Fig. 24, the header analyzer as described in the fourth embodiment can be added, as a header analysis 501, to the construction in Fig. 19 of the third embodiment. That is, as a result of frame header analysis, if it is determined that the frame has been encoded without interframe correlation, a code length comparator 502 determines objects to be combined, based on code lengths of respective objects, as in the case of the third embodiment. Information indicative of objects to be combined is held in the code length comparator 502, and only if an instruction is inputted

from the header analyzer 501, the held content is updated.

[0126] According to the constructions in Figs. 23 and 24, objects to be combined are re-determined upon decoding of frame encoded by the Intraframe-coding, whereby change of coding efficiency due to object combining can be suppressed.

[0127] Further, in the MPEG4 standard, handling of sound data as an object is studied. If a distance between sound sources of sound objects is regarded as a distance between objects, the first embodiment is applicable, and if the movement of sound source is object motion, the second embodiment is applicable. In use of code lengths of respective objects, the third embodiment is applicable. Thus, the above-described respective embodiments are applicable to coding of sound including audio information.

[0128] As described above, the first to fourth embodiments provide decoding apparatus and method which decode all the objects even if the number of decoders is limited.

[0129] Further, the embodiments provide decoding apparatus and method which decode a coded still image without degrading the image quality even if the number of decoders is limited.

[0130] Further, the embodiments provide decoding apparatus and method which decode a coded moving image without degrading the image quality even if the number of decoders is limited.

Fifth Embodiment

[Construction]

[0131] Fig. 29 is a block diagram showing the construction of a moving image processing apparatus according to a fifth embodiment of the present invention. In the present embodiment, the MPEG4 coding is used as a moving image coding method. Note that the coding method of the present embodiment is not limited to the MPEG4 coding, but any other coding method can be employed as long as it respectively encodes a plurality of objects within an image.

[0132] In Fig. 29, numeral 1201 denotes an encoder which inputs a moving image and encodes the image by the MPEG4 coding of Core profile and level 2. Numeral 1202 denotes a storage device used for storing coded moving image data. The storage device 1202 comprises a magnetic disk, an magneto-optic disk or the like. As the storage device 1202 is removably attached to the moving image processing apparatus, coded moving image data can be read in another apparatus. Numeral 1203 denotes a transmitter which transmits encoded moving image data to a LAN or a communication line, and performs broadcasting or the like; 1204, a receiver which receives code data outputted from the transmitter 1203; 1205, a profile and level regulator to which the present invention is applied; 1206, a storage device used for storing output from the profile and level regulator 1205; 1207, a decoder which decodes code data encoded by the MPEG4 coding of Core profile and level 1; and 1208, a display unit which displays a moving image decoded by the decoder 1207. Note that as described above, the encoder 1201 performs coding of Core profile and level 2, and in this example, to simplify the explanation, the encoder 1201 performs coding at a bit rate of 384 kbps.

[0133] Fig. 43 shows an example of an image to be encoded. In Fig. 43, respective numerals denote objects. An object 2000 represents background; an object 2001, a balloon moving in the air; an object 2002, a bird; objects 2003 and 2004, a woman and a man.

[0134] Fig. 31A shows a bit stream when the image in Fig. 43 is encoded. The bit stream has an arrangement information α indicative of location information of objects 2000 to 2004 at its head. Actually, the arrangement information α is encoded in BIFS (Binary Format for Scene description) language to describe scene construction information, and the arrangement information α is multiplexed. Then, VOSSC, Visual Object data α-1, α-2, α-3 and VOSEC follow. The code data in Fig. 31A is stored into the storage device 1202 or transmitted via the transmitter 1203. The code data is inputted via the storage device 1202 or the receiver 1204, into the profile and level regulator 1205 as a characteristic element of the present invention. The profile and level regulator 1205 also inputs the status of the decoder 1207.

[0135] Fig. 30 is a block diagram showing the detailed construction of the profile and level regulator 1205. In Fig. 30, numeral 1101 denotes the code data shown in Fig. 31A; 1102, a separator which separates the code data 1101 into code data indicative of arrangement information and header information, and code data indicative of respective objects; 1103, a header memory for storing code data indicative of separated arrangement information and header information; 1104 to 1108, code memories for storing code data for respective objects; 1109, a profile and level extractor which extracts the PLI code from the code data 1101, and extracts information on the profile and level; and 1110, an object counter which counts the number of objects included in the code data 1101.

[0136] Numeral 1111 denotes a decoder status receiver which obtains coding specifications (profile and level) of the decoder 1207 and other conditions; and 1112, a profile and level input unit through which arbitrary profile and level are set from a terminal (not shown) or the like; 1113, a profile and level determination unit which compares outputs from the profile and level extractor 1109 and the object counter 1110 with profile and level information inputted from the decoder status receiver 1111 or the profile and level input unit 1112, and determines whether or not the number of objects must be regulated.

[0137] Numeral 1114 denotes a code length comparator which determines the order of code lengths of objects by counting the code lengths of objects when the code data 1101 is inputted and comparing the code lengths with each other; 1115, a header changer which changes the content of header information stored in the header memory 1103, based on the outputs from profile and level determination unit 1113 and the code length comparator 1114; 1116, a multiplexer which multiplexes code data read from the code memories 1104 to 1108 based on the output from the header changer 1115 and the results of comparison by the code length comparator 1114; and 1117, code data outputted as a result of profile and level regulation.

[Regulation of Profile and Level]

[0138] Hereinbelow, the processing in the profile and level regulator 1205 having the above construction will be described in detail.

[0139] The code data 1101 is inputted into the separator 1102, the profile and level extractor 1109, the object counter 1110 and the code length comparator 1114. The separator 1102 separates the code data 1101 into code data indicative of arrangement information and header information, and code data indicative of respective objects, and stores the respective code data into the header memory 1103 and the code memories 1104 to 1108. For example, the object arrangement information α, VOSSC, Visual Object SC, the respective code immediately prior to the VO data A, and the header information of VOL and VOP data in Fig. 25, and the like, are stored in the header memory 1103. Further, the VOL and VOP data for the respective object, where the header information is removed, are stored in the code memories 1104 to 1108. These data are stored independently such that the header-removed part is clearly indicated. For example, in the image in Fig. 43, as the number of objects is five, the code data of the objects 2000 to 2004 (VO data A to E in Fig. 31A) are respectively stored into the code memories 1104 to 1108.

[0140] At the same time, the object counter 1110 counts the number of objects included in the code data 1101. Then the code length comparator 1114 measures code lengths of the respective objects.

[0141] The profile and level extractor 1109 extracts PLI-α from the code data 1101 and decodes it, to extract information on the profile and level of the code data 1101. At the same time of extraction, the decoder status receiver 1111 operates, to obtain information on the profile, level and the like, decodable by the decoder 1207. These information may be set by the user via the profile and level input unit 1112.

[0142] The profile and level determination unit 1113 compares the profile and level information, obtained from the decoder 1207, or set by the user, with the result of extraction by the profile and level extractor 1109. If the obtained or set profile and level are higher than or equal to those extracted from the code data 1101, the profile and level determination unit 1113 does not operate the header changer 1115. Then, the contents of the header memory 1103 and the code memories 1104 to 1108 are read in the order of input, and multiplexed by the multiplexer 1116. Thus, code data 1117 is generated. That is, the contents of the code data 1117 are the same as that of the code data 1101.

[0143] On the other hand, if the profile and level obtained from the decoder 1207 or set by the user are lower than the profile and level extracted from the code data 1101, the profile and level determination unit 1113 inputs the number of objects included in the code data 1101 from the object counter 1110, and compares the number of objects with the number of decodable objects, determined from the obtained or set profile and level information.

[0144] If the number of objects obtained by the object counter 1110 is less than the number of decodable objects, the code data 1117 is generated, as in the case of the above-described case where the obtained or set profile and level are higher than or equal to those extracted from the code data 1101.

[0145] On the other hand, if the number of objects obtained by the object counter 1110 is greater than the number of decodable objects, the number of decodable objects is inputted into the code length comparator 1114, and the code lengths are compared with each other. The code length comparator 1114 sets objects to be decoded, from an object having the longest code length. That is, the objects are decoded sequentially from the object having the longest code length. For example, in Fig. 31A, if the code length of video object becomes shorter, in the order in which the VO data A, the VO data D, the VO data C, the VO data E, and the VO data B appear, as the decoder 1207 performs decoding of Core profile and level 1, it can decode to a maximum of four objects. Accordingly, the code length comparator 1114 disables reading of the VO data B from the code memory 1106, and enables reading from the code memories 1104, 1105, 1107 and 1108.

[0146] Then the profile and level determination unit 1113 operates the header changer 1115 to change the content of PLI in correspondence with the decoder 1207, then, coding is performed. In this manner, header information on undecodable (deleted) object (VO data B in this case) by the decoder 1207 is deleted, based on the result of comparison by the code length comparator 1114. That is, the header information of the code data 1101 is rewritten with contents corresponding to the decoding capability of the decoder 1207 or the set profile and level. Further, arrangement information on the object 2002 corresponding to the deleted object (VO data B) is deleted from the arrangement information α, and new arrangement information β is generated.

[0147] Then, the contents of the header changer 1115 and the code memories 1104, 1105, 1107 and 1108 are read

in the order of input, and multiplexed by the multiplexer 1116, thus the code data 1117 is generated. Fig. 31B shows a bit stream of the code data 1117. In Fig. 31B, the newly generated arrangement information β is provided at the head of the bit stream, then, VOSSC, Visual Object data β-1, β-2, β-3, and VOSEC follow. The Visual Object data β-1, β-2 and β-3 are obtained by regulating the number of objects with respect to the original Visual Object data α-1, α-2 and α-3 in Fig. 31A. For example, the Visual Object data β-1 comprises the Visual Object SC positioned at the head, PLI-β indicative of the profile and level corresponding to the decoder 1207, and code data where the code data (VO data B) on the object 2002 is deleted.

[0148] The code data 1117 obtained as above is stored into the storage device 1206, or decoded by the decoder 1207 and displayed on the display unit 1208. Fig. 44 shows a displayed image, represented by the decoded code data 1117. In Fig. 44, the object 2002, representing the bird in the image as the object of encoding in Fig. 43, is deleted.

[0149] Note that in the above description, the code length comparator 1114 directly counts the code lengths from the code data 1101, however, the code length comparator 1114 may count the code lengths based on the code data stored in the code memories 1104 to 1108.

[0150] As described above, according to the present embodiment, even if the coding specifications (profile and/or level) of a decoder are different from those of an encoder, code data can be decoded. Further, by deleting object data having the shortest code length, selection of object to be deleted is facilitated, and the influence on a decoded image can be suppressed as much as possible.

[0151] Further, even if the number of objects decodable by the decoder 1207 is less than the number defined by the coding specifications of the code data 1101, as the decoder status receiver 1111 obtains the number of actually decodable objects, similar advantages can be attained.

[0152] In addition, even when code data having coding specifications higher than or equal to those of the decoder 1207 is inputted, by deleting object(s) to reduce the bit rate, decoding by the decoder 1207 can be performed.

Sixth Embodiment

[0153] Hereinbelow, a sixth embodiment of the present invention will be described. Note that the general construction of the moving image processing apparatus according to the sixth embodiment is similar to that in Fig. 29 of the above-described fifth embodiment, therefore, an explanation of the construction will be omitted.

[0154] Fig. 32 is a block diagram showing the construction of the profile and level regulator 1205 according to the sixth embodiment of the present invention. In Fig. 32, elements corresponding to those in Fig. 30 have the same reference numerals and explanations of the elements will be omitted. In the sixth embodiment, the MPEG4 coding is employed as a moving image coding method, however, any other coding method is applicable as long as it encodes a plurality of objects within an image

[0155] In Fig. 32, numeral 1118 denotes a size comparator which extracts sizes of respective objects from the header memory 1103 and compares the sizes with each other.

[0156] As in the case of the fifth embodiment, the code data 1101 is inputted into the separator 1102, the profile and level extractor 1109, the object counter 1110 and the code length comparator 1114, and the respective code data are stored into the header memory 1103 and the code memories 1104 to 1108. At the same time, the object counter 1110 counts the number of objects included in the code data.

[0157] The size comparator 1118 extracts an image size of each object, by extracting the respective VOL_width and VOL_height code in the bit stream structure in Fig. 25 and decoding the extracted codes.

[0158] Then, as in the case of the fifth embodiment, the profile and level extractor 1109 extracts information on the profile and level from the code data 1101, and at the same time, information on profile and level and the like of the decoder 1207 is obtained from the decoder status receiver 1111, or the profile and level are set by the user from the profile and level input unit 1112.

[0159] The profile and level determination unit 1113 compares the profile and level information obtained from the decoder 1207 or set by the user, as described above, with the result of extraction by the profile and level extractor 1109. If the obtained or set profile and level are higher than or equal to the profile and level extracted from the code data 1101, profile and level determination unit 1113 does not operate the header changer 1115. Then, the code data 1117 similar to the code data 1101 is generated.

[0160] On the other hand, if the profile and level obtained from the decoder 207 or set by the user are lower than the profile and level extracted from the code data 1101, the profile and level determination unit 1113 inputs the number of objects included in the code data 1101 from the object counter 1110, and compares the input number with the number of decodable objects determined from the obtained or set profile and level.

[0161] Then, if the number of objects obtained by the object counter 1110 is less than the number of decodable objects, the code data 1117 is generated as in the above-described case where the obtained or set profile and level are higher than or equal to those of the code data 1101.

[0162] On the other hand, if the number of objects obtained by the object counter 1110 is greater than the number of

decodable objects, the number of decodable objects is inputted into the size comparator 1118, and size comparison is performed. The size comparator 1118 sets a plurality of objects of the code data 1101, sequentially from the largest image size, as objects to be decoded. That is, the objects are decodable, sequentially from the largest image size. For example, in Fig. 43, in the image sizes of the respective objects, the image size becomes smaller in the order in which the objects 2000, 2004, 2001, 2003 and 2002 appear. As the decoder 1207 performs decoding Core profile and level 1, it can decode to a maximum of four objects. Accordingly, in the image in Fig. 43, except the smallest object 2002, the other four objects can be decoded. The size comparator 1118 disables reading of the code data of the object 2002 from the code memory 1106, and enables reading from the code memories 1104, 1105, 1107 and 1108.

[0163] Then, as in the case of the fifth embodiment, the profile and level determination unit 1113 operates the header changer 1115 to change the content of PLI in correspondence with the decoder 1207, then, coding is performed. Further, header information on the undecodable (deleted) object (object 2002 in this case) by the decoder 1207 is deleted, based on the result of comparison by the size comparator 1118. Further, arrangement information on the deleted object 2002 is deleted from the arrangement information α, and new arrangement information β is generated.

[0164] Then, the contents of the header changer 1115 and the code memories 1104, 1105, 1107 and 1108 are read in the order of input, and multiplexed by the multiplexer 1116, thus the code data 1117 is generated. Fig. 31B shows a bit stream of the code data 1117 at this time.

[0165] The code data 1117 obtained as above is stored into the storage device 1206, or decoded by the decoder 1207 and displayed, as an image as shown in Fig. 44, on the display unit 1208.

[0166] Note that in the above description, the size comparator 1118 extracts image sizes of objects based on the VOL_width and VOL_height code of the code data 1101, however, the size comparator 1118 may extract the image sizes based on the VOP_width and VOP_height code, or based on shape (mask) information obtained by decoding code data indicative of shape (mask) information.

[0167] As described above, according to the sixth embodiment, even if the coding specifications of a decoder are different from those of an encoder, code data can be decoded. Further, by deleting object data having the minimum image size, selection of object to be deleted is facilitated, and the influence on a decoded image can be suppressed as much as possible.

[0168] Note that in the fifth and sixth embodiments, only one object is deleted, however, two or more object can be deleted. Further, it may be arranged such that the user directly designates object(s) to be deleted.

[0169] Further, it may be arranged such that the order of deletion is set for the respective objects of image in advance by the profile and level input unit 1112.

Seventh Embodiment

[0170] Hereinbelow, a seventh embodiment of the present invention will be described. Note that the general construction of the moving image processing apparatus according to the seventh embodiment is similar to that in Fig. 29 of the fifth embodiment, therefore, an explanation of the construction will be omitted.

[0171] Fig. 33 is a block diagram showing the detailed construction of the profile and level regulator 1205 according to the seventh embodiment of the present invention. In Fig. 33, elements corresponding to those in Fig. 30 have the same reference numerals and explanations of the elements will be omitted. In the seventh embodiment, the MPEG4 coding is employed as a moving image coding method, however, any other coding method is applicable as long as it encodes a plurality of objects within an image.

[0172] In Fig. 33, numeral 1120 denotes an object selection designator which displays a plurality of objects, and in which the user's designation of arbitrarily selected objects is inputted; 1121, an object selector which selects code data of objects to be processed, based on designation from the object selection designator 1120, and the result of determination by the profile and level determination 1113; 1122 and 1124, selectors, controlled by the object selector 1121, which switch their input and output; and 1123, an object integrator which integrates a plurality of objects.

[0173] As in the case of the above-described fifth embodiment, the code data 1101 is inputted into the separator 1102, the profile and level extractor 1109 and the object counter 1110. The separator 1102 separates the code data 1101 into code data indicative of arrangement information and header information and code data indicative of respective objects. The respective code data are stored into the header memory 1103 and the code memories 1104 to 1108. At the same time, the object counter 1110 counts the number of objects included in the code data 1101.

[0174] Then, as in the case of the fifth embodiment, the profile and level extractor 1109 extracts information on the profile and level from the code data 1101. The decoder status receiver 1111 obtains information on profile and level and the like of the decoder 1207. Further, the profile and level are set by the user via the profile and level input unit 1112.

[0175] The profile and level determination unit 1113 compares the profile and level information obtained from the decoder 1207 or set by the user, as described above, with the result of extraction by the profile and level extractor 1109. If the obtained or set profile and level are higher than or equal to the profile and level extracted from the code data 1101, profile and level determination unit 1113 controls the object selector 1121 to select a path directly connecting the selec-

tor 1122 to the selector 1124 such that the code data does not pass through the object integrator 1123. The header changer 1115 is not operated. The code data stored in the header memory 1103 and the code memories 1104 to 1108 are read out in the order of input, and multiplexed by the multiplexer 1116. Thus, the code data 1117 similar to the code data 1101 is generated.

[0176] On the other hand, if the profile and level obtained from the decoder 1207 or set by the user are lower than the profile and level extracted from the code data 1101, the profile and level determination unit 1113 inputs the number of objects included in the code data 1101 from the object counter 1110, and compares the number of objects with the number of decodable objects, determined from the obtained or set profile and level information.

[0177] If the number of objects obtained by the object counter 1110 is less than the number of decodable objects, the code data 1117 is generated, as in the case of the above-described case where the obtained or set profile and level are higher than or equal to those extracted from the code data 1101.

[0178] On the other hand, if the number of objects obtained by the object counter 1110 is greater than the number of decodable objects, the number of decodable objects is inputted into the object selector 1121. The object selector 1121 displays statuses of the respective objects (e.g., the image in Fig. 43), information on the respective objects, information on the number of integrated objects and the like, on the object selection designator 1120. The user selects objects to be integrated in accordance with these information, and inputs an instruction on the selection into the object selection designator 1120.

[0179] In the seventh embodiment, as the decoder 1207 performs decoding of Core profile and level 1, it can decode to a maximum of four objects. For example, as the image in Fig. 43 has five objects, two of them are integrated into one object, whereby code data decodable by the decoder 1207 can be obtained. Hereinbelow, a case where the user designated integration of the object 2003 and the object 2004 in the image in Fig. 43 will be described.

[0180] When the user designates the objects to be integrated via the object selection designator 1120, the profile and level determination unit 1113 operates the header changer 1115 to change the content of PLI in correspondence with the decoder 1207, generate header information on the new object obtained by integration and delete header information on the objects deleted by the integration, based on the result of selection by the object selector 1121. More specifically, arrangement information of the new object obtained as a result of integration is generated and arrangement information of the original objects 2003 and 2004 are deleted, based on the arrangement information of the objects 2003 and 2004. Then, the size of the object obtained by the integration or other information are generated as header information and header information of the original objects 2003 and 2004 are deleted, based on the header information of the objects 2003 and 2004.

[0181] The object selector 1121 controls the input/output of the selectors 1122 and 1124 so as to perform integration processing by the object integrator 1123 with respect to code data of the objects 2003 and 2004, and to avoid processing by the object integrator 1123 with respect to other code data.

[0182] Then, contents of the header changer 1115 and the code memories 1104 to 1106 holding the code data of the objects 2000 to 2002 are read out in the order of input, and multiplexed by the multiplexer 1116 via the selectors 1122 and 1124. On the other hand, the contents of the code memories 1107 and 1108 holding the code data of the objects 2003 and 2004 to be integrated are inputted via the selector 1122 to the object integrator 1123.

[Object Integrator]

[0183] Fig. 34 is a block diagram showing the detailed construction of the object integrator 1123. In Fig. 34, numerals 1050 and 1051 denote code memories respectively for storing code data of objects to be integrated; 1052 and 1054, selectors which switch input/output for respective objects; 1053, an object decoder which decode code data and reproduces an image of an object; 1055 and 1056, frame memories for storing reproduced images for respective objects; 1057, a synthesizer which synthesizes objects in accordance with arrangement information of objects to be integrated stored in the header memory 1103; and 1058, an object encoder which encodes image data obtained by synthesizing and outputs the image data.

[0184] Hereinbelow, the operation of the object integrator 1123 will be described in detail. The code data of the objects 2003 and 2004 to be integrated are stored into the code memories 1050 and 1051. First, the selector 1052 selects an input on the code memory 1050 side, and the selector 1054, an output on the frame memory 1055 side. Thereafter, the code data is read out from the code memory 1050, and decoded by the object decoder 1053. Then image information of the object 2003 is written via the selector 1054 into the frame memory 1055. The image information of the object comprises image data indicative of a color image and mask information indicative of a shape. Then, the input and output of the selector 1052 and 1054 are switched to the opposite sides, and similar processing is performed, whereby the image information of the object 2004 is stored into the frame memory 1056.

[0185] The synthesizer 1057 obtains location information and size information of the objects 2003 and 2004 from the header memory 1103, and obtains the image size of the new object obtained by object synthesizing and relative locations of the original objects 2003 and 2004 in the new object. Then, the image information in the frame memories 1055

and 1056 are read out, and the color image information and the mask information are respectively synthesized. Fig. 37 is shows the result of synthesizing of color image information. Fig. 38 shows the result of synthesizing of mask information. The object encoder 1058 encodes these color image information and mask information in accordance with the MPEG4 object coding. Then, the object integrator 1123 outputs the encoded information.

[0186] The code data outputted from the object integrator 1123 is multiplexed with other code data by the multiplexer 1116 via the selector 1124, thus the code data 1117 is obtained. Fig. 35 shows a bit stream of the code data 1117. Fig. 35 shows the result of integration processing according to the seventh embodiment with respect to the code data 1101 in Fig. 31A. In Fig. 35, the bit stream has arrangement information γ including arrangement information of the newly obtained object as the result of synthesizing, VOSSC, Visual Object data γ-1, γ-2 and γ-3, and VOSEC. The Visual Object data γ-1, γ-2 and γ-3 are obtained by object integration regulation with respect to the original Visual Object data α-1, α-2 and α-3 shown in Fig. 31A. For example, the Visual Object data γ-1, following Visual Object SC, comprises PLI-γ indicative of profile and level appropriate to the decoder 1207, VO data A, VO data B and VO data C as respective code data of the objects 2000 to 2002, and code data VO data G obtained by integrating the objects 2003 and 2004.

[0187] The code data 1117 obtained as above is stored into the storage device 1206, or decoded by the decoder 1207 and reproduced as an image as shown in Fig. 43 and displayed on the display unit 1208.

[0188] Note that in the seventh embodiment, the user selects and designates objects to be integrated within an image by the object selection designator 1120, however, the present invention is not limited to this example. For example, it may be arranged such that the integration order is set for objects of the image in advance by the object selection designator 1120, then if the number of objects decodable by the decoder 1207 is less than the number of objects of the image and object integration is required, object integration is automatically performed in accordance with the set order.

[0189] As described above, according to the seventh embodiment, even if profile and/or level of a decoder are different from those of an encoder, code data can be decoded. Further, by integrating objects and decoding the integrated object, loss of decoded object can be prevented.

[0190] Further, the object integration processing can be performed in incremental order of code length or image size by providing the code length comparator 1114 and the size comparator 1118 shown in the fifth and sixth embodiments in place of the object selection designator 1120 and the object selector 1121 for controlling the object integrator 1123.

[0191] Fig. 36 is a block diagram showing the construction of the object integrator 1123 according to a modification of the seventh embodiment. In Fig. 36, elements corresponding to those in Fig. 34 have the same reference numerals and explanations of the elements will be omitted. The construction of Fig. 36 is characterized by further comprising a code length counter 1059. The code length counter 1059 counts code lengths of code data of respective objects prior to integration, and parameters (e.g., quantization parameters or the like) of the object encoder 1058 is controlled such that the code length of output from the object encoder 1058 is the same as the counted result. Thus the objects can be synthesized without increasing the total code length.

Eighth Embodiment

[0192] Hereinbelow, an eighth embodiment of the present invention will be described. As in the case of the above-described seventh embodiment, object integration processing is performed in the eighth embodiment. Note that the general construction of the moving image processing apparatus of the eighth embodiment, and the detailed construction of the profile and level regulator 1205 are the same as those in Fig. 33, therefore explanations of the apparatus and the construction of the profile and level regulator will be omitted.

[0193] Fig. 39 is a block diagram showing the detailed construction of the object integrator 1123 according to the eighth embodiment of the present invention. In Fig. 39, elements corresponding to those in Fig. 34 have the same reference numerals and explanations of the elements will be omitted.

[0194] In Fig. 39, numerals 1060 and 1061 denote separators which separate input code data into code data on mask information indicative of shape and code data indicative of color image information and output the separated data. Numeral 1062 to 1065 denote code memories. The code data indicative of color image information is stored into the code memories 1062 and 1064, and the code data on mask information is stored into the code memories 1063 and 1065, for respective objects. Numeral 1066 denotes a color image information code synthesizer which synthesizes the code data indicative of color image information in the form of code data; 1067, a mask information code synthesizer which synthesizes the code data indicative of mask information in the form of code data; 1068, a multiplexer which multiplexes code outputted from the color image information code synthesizer 1066 and the mask information code synthesizer 1067.

[0195] Hereinbelow, the operation of object integrator 1123 according to the eighth embodiment will be described in detail. As in the case of the seventh embodiment, the code data of the objects 2003 and 2004 are stored respectively into the code memories 1050 and 1051. The code data of the object 2003 stored in the code memory 1050 is read out in frame units (VOP units), separated by the separator 1060 into code data of color image information and code data of mask information, and the respective code data are stored into the code memories 1062 and 1063. Similarly, code data

of color image information and code data of mask information of the object 2004 are stored into the code memories 1064 and 1065.

[0196] Thereafter, the color image information code synthesizer 1066 reads the color image information code data from the code memories 1062 and 1064. Further, as in the case of the seventh embodiment, the color image information code synthesizer 1066 obtains location information and size information of the objects 2003 and 2004 from the header memory 1103, and obtains the image size of synthesized new object and respective relative locations of the original objects 2003 and 2004 in the new object. That is, the color image information code synthesizer 1066 performs synthesizing on the assumption that if these color image information code data are synthesized and decoded, an image as shown in Fig. 37 can be obtained as one object.

[0197] Note that the MPEG4 coding method has a slice data structure to define a plurality of macro blocks as a cluster of blocks in a main scanning direction. Fig. 40 shows an example of the slice structure applied to the objects in Fig. 37. In Fig. 40, an area in a bold frame is defined as one slice. In each slice, the head macro block is hatched.

[0198] The color image information code synthesizer 1066 performs reading in a rightward direction (main scanning direction) as shown in Fig. 40, sequentially from an upper left macro block data of the image, to be obtained as a result of synthesizing. That is, among the code data of the object 2003, code data corresponding to the head macro block of the head slice is read from the code memory 1062 first. The header information of the slice is added to the read code data, and the code data of the head macro block is outputted. Then, the code data corresponding to the macro block on the right of the head macro block is read and outputted. In this manner, the read and output operations are sequentially repeated to the slice.

[0199] Note that a portion where data has been newly generated between the objects 2003 and 2004 is considered as a new slice. As this portion is not displayed even if decoded with mask information, appropriate pixels are provided to cover the portion. That is, such portion comprises only DC component of the last macro block including an object. As the DC difference is "0", and all the AC coefficients are "0", no code is generated.

[0200] Then, as it is considered that a new slice has started on the edge of the object 2004, a hatched macro block in Fig. 40 is regarded as the head of new slice, and the header information of the slice is added to the block. In this case, as the address of the head macro block is an absolute address, the address is converted to a relative address from the macro block including the previous object. Note that in the macro block, if DC component or the like is predicted by referring to another macro block, that portion is re-encoded, then code data of the macro block is sequentially outputted in the rightward direction. That is, the slice header is added on the edge of object, and the prediction of the slice head macro block is replaced with initialized code. The obtained code is outputted to the multiplexer 1068.

[0201] In parallel to the operation of the color image information code synthesizer 1066, the mask information code synthesizer 1067 reads the code data of the mask information from the code memories 1063 and 1065. Then, the mask information code synthesizer 1067 obtains location information and size information of the objects 2003 and 2004 from the header memory 1103, and obtains the image size of a synthesized new object and relative locations of the original objects 2003 and 2004 in the new object. Then, by decoding and synthesizing the input code data of the mask information, the mask information code synthesizer 1067 obtains mask information as shown in Fig. 38. The mask information code synthesizer 1067 encodes the mask image by an arithmetic encoding as the MPEG4 shape information coding method. The obtained code is outputted to the multiplexer 1068.

[0202] Note that the mask information coding is not limited to the MPEG4 arithmetic coding method. For example, in the result of synthesizing of mask information code data, as the zero-run between object edges is merely lengthened, the synthesizing can be made only by replacing code representing the zero-run length without decoding by the mask information code synthesizer 1067, by employing zero-run coding or the like, used in a facsimile apparatus. Generally, even when mask information is encoded by the arithmetic or another coding, the code length is merely slightly changed.

[0203] The multiplexer 1068 multiplexes the code data on the synthesized color image information and the code data of the mask information, as code data of one object. The subsequent processing is similar to that in the above-described seventh embodiment. The multiplexer 1116 multiplexes the code data with other code data and outputs the data.

[0204] As described above, according to the eighth embodiment, even in a case where the profile and/or level of an encoder are different from those of a decoder, code data can be decoded. Further, as objects are integrated in the form of code data, loss of object in decoded image data can be prevented only by adding header information.

[0205] Further, in the object integration processing according to the eighth embodiment, a newly added header can be obtained by a slight amount of calculation, and further, code change is limited to the head block of a slice. Accordingly, the object integration processing can be performed at a speed higher than that in the object integration processing by decoding and re-encoding according to the seventh embodiment.

Ninth Embodiment

[0206] Hereinbelow, a ninth embodiment of the present invention will be described. In the ninth embodiment, object

integration processing is performed, as in the case of the above-described seventh embodiment,. Note that the general construction of the moving image processing apparatus according to the ninth embodiment is similar to that in Fig. 29 of the fifth embodiment, therefore, an explanation of the construction will be omitted.

[0207] Fig. 41 is a block diagram showing the detailed construction of the profile and level regulator 1205 according to the ninth embodiment of the present invention. In Fig. 41, elements corresponding to those of the seventh embodiment in Fig. 33 have the same reference numerals, and explanations of the elements will be omitted. In the ninth embodiment, the MPEG4 coding is employed as a moving image coding method, however, any other coding method is applicable as long as it encodes a plurality of objects within an image.

[0208] In Fig. 41, numeral 1170 denotes an object arrangement information determination unit which determines objects to be integrated.

[0209] As in the case of the seventh embodiment, the profile and level determination unit 1113 compares the profile and level information of the decoder 1207 with those of the code data 1101. Even if the profile and level of the decoder 1207 are higher than or equal to, or lower than those of the code data 1101, the code data 1117 is generated in a similar manner to that of the seventh embodiment as long as the number of objects obtained by the object counter 1110 is the number decodable by the decoder 1207.

[0210] On the other hand, if the number of objects obtained by the object counter 1110 is greater than the number decodable by the decoder 1207, the decodable number of objects is inputted into the object arrangement information determination unit 1170. As in the case of the seventh embodiment, the maximum number of objects decodable by the decoder 1207 is four. Accordingly, in an image having five objects as shown in Fig. 43, decodable code data can he obtained by integrating two objects.

[0211] The object arrangement information determination unit 1170 extracts location information and size information of the respective objects from the header memory 1103, and determines two objects to be integrated based on the following conditions. Note that condition (1) is given higher priority to condition (2).

(1) One object is included in the other object
(2) The distance between both objects is the shortest

[0212] In the image shown in Fig. 43, the objects 2001 to 2004 are included in the object 2000. Accordingly, the object arrangement information determination unit 1170 determines the object 2000 and the object 2001 as objects to be integrated.

[0213] When the objects to be integrated have been determined, the profile and level determination unit 1113 operates the header changer 1115 to change and encode the content of the PLI in accordance with the decoder 1207, and generate header information on a new object obtained by object integration and delete the header information on the integrated objects, as in the case of the seventh embodiment, based on the result of determination by the object arrangement information determination unit 1170. More specifically, arrangement information on the new object obtained by object integration is generated, based on the arrangement information of the objects 2000 and 2001, and arrangement information of the original objects 2000 and 2001 are deleted. Then, the image size information or other information of the object obtained by the integration is generated as header information, based on the header information of the objects 2000 and 2001, and the header information of the original objects 2000 and 2001 are deleted.

[0214] The object arrangement information determination unit 1170 controls input/output of the selectors 1122 and 1124 so as to perform integration processing on the code data of the objects 2000 and 2001 by the object integrator 1123, on the other hand, so as not to pass the other code data through the object integrator 1123.

[0215] Then, the contents of the header changer 1115 and the code memories 1106 to 1108 holding the code data of the objects 2002 to 2004 are read out sequentially in the order of input, and inputted via the selectors 1122 and 1124 into the multiplexer 1116. On the other hand, the contents of the code memories 1104 and 1105 holding the code data of the objects 2000 and 2001 to be integrated are integrated by the object integrator 1123, and inputted into the multiplexer 1116. The multiplexer 1116 multiplexes these code data, thus generates the code data 1117. Note that the integration processing by the object integrator 1123 is realized in a similar manner to that in the above-described seventh embodiment or eighth embodiment.

[0216] Fig. 42 shows a bit stream of the code data 1117 according to the ninth embodiment. Fig. 42 shows the result of integration processing of the ninth embodiment performed on the code data 1101 as shown in Fig. 31A. In Fig. 42, arrangement information δ including arrangement information of the newly obtained object is provided at the head. Then VOSSC, Visual Object data δ-1, δ-2, δ-3, and VOSEC follow. The Visual Object data δ-1, δ-2, δ-3 are obtained by performing object integration regulation on the original Visual Object data α-1, α-2, and α-3 in Fig. 31A. For example, the Visual Object data δ-1 comprises Visual Object SC, then PLI-δ indicative of profile and level appropriate to the decoder 1207, VO data H as code data obtained by integrating the objects 2000 and 2001, and VO data C, VO data D and VO data E as code data of the objects 2002 to 2004.

[0217] The code data 1117 obtained as above is stored into the storage device 1206, or decoded by the decoder 1207

and reproduced as an image as shown in Fig. 43, and displayed on the display unit 1208.

[0218] Note that in the ninth embodiment, as in the cases of the fifth and sixth embodiments, code lengths of respective objects, object sizes and the like may be added to the conditions for determining objects to be integrated.

[0219] As described above, according to the ninth embodiment, even if the profile and/or level of an encoder are different from those of decoder, code data can be decoded. Further, loss of decoded object can be prevented while suppressing the amount of code changed by integration, by integrating the objects based on the location relation among the objects.

[0220] In the ninth embodiment, objects to be integrated are determined based on the location relation among the objects. The determination according to the ninth embodiment may be employed in the above-described fifth and sixth embodiments. That is, objects to be deleted can be selected based on location information of objects.

[0221] Note that in the seventh to ninth embodiments, two objects are integrated and one object is generated. However, three or more objects, or two or more sets of objects may be integrated.

[0222] Note that the arrangement of the code memories 1104 to 1108 and the header memory 1103 is not limited to that shown in Fig. 41. More code memories can be provided, or one memory may be divided into a plurality of areas. Further, a storage medium such as a magnetic disk may be employed.

[0223] Further, the selection of objects to be deleted or integrated may be determined based on the combination of a plurality of conditions such as sizes and code lengths of objects, location relation among the objects and user's instruction.

[0224] Further, in a case where the fifth to ninth embodiments are applied to an image editing apparatus, even if the number of objects changes due to editing processing, the output from the apparatus can be adjusted to an arbitrary profile and/or level.

[0225] As described above, according to the fifth to ninth embodiments, code data encoded for a plurality of image information (objects) can be decoded by decoders of arbitrary specifications. Further, the number of objects included in the code data can be regulated.

Tenth Embodiment

[Construction]

[0226] Fig. 45 is a block diagram showing the construction of the moving image processing apparatus according to a tenth embodiment of the present invention. In the tenth embodiment, the MPEG4 coding is employed as a moving image coding method. Note that the coding method is not limited to the MPEG4 coding, but any other coding method is applicable as long as it encodes a plurality of objects within an image.

[0227] In Fig. 45, numerals 2201 and 2202 denote storage devices holding moving image code data. The storage devices 2201 and 2202 respectively comprise a magnetic disk, an magneto-optical disk, a magnetic tape, a semiconductor memory or the like. Numeral 2203 denotes a TV camera which obtains a moving image and outputs a digital image signal; 2204, an encoder which performs coding by the MPEG4 coding method; 2205, a communication line of a local area network (LAN), a public line, a broadcasting line or the like; 2206, a communication interface which receives coded data from the communication line 2205; and 2207, an editing operation unit which displays image editing condition. The user inputs editing instruction from the editing operation unit 2207. Further, numeral 2208 denotes an image editing unit characteristic of the present embodiment; 2209, a storage device for storing output from the image editing unit 2208; 2210, a decoder which decodes code data of a moving image encoded by the MPEG4 coding; 2211, a display unit which displays a moving image decoded by the decoder 2210.

[Image Editing]

[0228] Hereinbelow, image editing processing of the present embodiment will be described using a specific image as an example.

[0229] Image data, encoded by the MPEG4 coding of Core profile and level 2 at a bit rate of 384 kbps, is stored into the storage device 2201. Fig. 46A shows an example of the image stored in the storage device 2201. Fig. 50A shows the code data of the image. In the image of Fig. 46A, a background object 2300 includes objects 2304 and 2305 representing men. In Fig. 50A, code data of the background object 2300 is VO data A-1-1, and code data of the men objects 2304 and 2305 are VO data A-1-2 and VO data A-1-3.

[0230] Image data, encoded by the MPEG4 coding of Core profile and level 1 at a bit rate of 200 kbps, is stored into the storage device 2202. Fig. 46B shows an example of the image stored in the storage device 2202. Fig. 50B shows the code data of the image. In the image of Fig. 46B, a background object 2301 includes objects 2306 and 2307 representing a man and a woman. In Fig. 50B, code data of the background object 2301 is VO data B-1-1, and code data of the man and woman objects 2306 and 2307 are VO data B-1-2 and VO data B-1-3.

[0231] In a case where the TV camera 2203 obtains an image as shown in Fig. 46C and the encoder 2204 encodes the image data by the MPEG4 coding of Simple profile and level 1 at a bit rate of 32 kbps, as a new object is not extracted from the obtained image, the entire image is handled as one object 2302. Accordingly, as shown in Fig. 50C, code data of the image comprises VO data C-1-1, code data of one object 2302.

[0232] Further, in a case where an image as shown in Fig. 46D is encoded by the MPEG4 coding of Simple profile and level 2 and inputted from the communication line 2205 via the communication interface 2206, a background object 2303 in the image in Fig. 46D includes objects 2308 and 2309 representing a woman and a man. Fig. 50D shows code data of the image, in which code data of the background object 2303 is VO data D-1-1, code data of the man and woman objects 2308 and 2309 are VO data D-1-2 and VO data D-1-3.

[0233] Note that to simplify the explanation, the sizes of all the above-described images (Figs. 46A to 46D) are defined with QCIF (Quarter Common Intermediate Format).

[Image Editing Unit]

[0234] All the code data are inputted into the image editing unit 2208. Fig. 47 is a block diagram showing the construction of the image editing unit 2208. In Fig. 47, numerals 2101 to 2104 denote system code memories for storing system-related code data for respective inputs; 2105 to 2108, video code memories for storing moving image code data for respective inputs; 2109, a video decoder which decodes moving image code data to reproduce objects; and 2110, a system decoder which decodes the system code data to reproduce object arrangement information and the like.

[0235] The results of decoding are outputted to the editing operation unit 2207, and the respective objects are displayed in accordance with the arrangement information. The editing operation unit 2207 newly sets display timing, speed and the like, in accordance with designation of arrangement of these objects, size change, deformation and the like, instructed by the user.

[0236] Numeral 2111 denotes a system code synthesizer which synthesizes system code; 2112, a header processor which synthesizes or changes headers of video code; 2113, a selector which arbitrarily selects one of outputs from the video code memories 2105 to 2108 and outputs the selected output; 2114, a multiplexer which multiplexes outputs from the system code synthesizer 2111, the header processor 2112 and the selector 2113 to generate code data.

[0237] In the image editing unit 2208, respective outputs from the storage devices 2201 and 2202, the encoder 2204 and the communication interface 2206 are separated into system code data and moving image code data. The system code data are stored into the system code memories 2101 to 2104, and the moving image code data are stored into the video code memories 2105 to 2108.

[0238] When the respective code data have been stored, the video decoder 2109 and the system decoder 2110 decode the respective data, and output the decoded data to the editing operation unit 2207. In the editing operation unit 2207, the user sets settings of deletion/holding objects, change of arrangement, moving image start timing, frame rate and the like. The video decoder 2209 and the system decoder 2110 arbitrarily perform decoding in accordance with the editing operation.

[0239] Fig. 48 shows an example of an image synthesized from the images shown in Figs. 46A to 46D. That is, a new image 2320 is generated by editing and synthesizing the four images. The size of the image 2320 is defined with CIF format because the QCIF four images are synthesized without overlapping with each other. In the image 2320, the background object 2300, the object 2302, the background objects 2303 and 2301 are arranged, from an upper left position in a clockwise manner. Further, the men objects 2304 and 2305 are moved horizontally in rightward direction (edited). The object 2308 is enlarged and moved onto the background object 2300 (edited).

[0240] The system code synthesizer 2111 reads out the system code data from the system code memories in accordance with the results of synthesizing, then generates new system code data with arrangement information corresponding to these deformation and movement, and outputs the new system code data to the multiplexer 2114.

[0241] Next, the changing condition accompanying synthesizing of respective objects will be described below.

[0242] First, regarding the background object 2300, coordinates, start timing and the like have not been changed. Regarding the background object 2301, its coordinates (0,0) has been changed to (0,144). Regarding the object 2302, its coordinates (0,0) has been changed to (176,0). Regarding the background object 2303, its coordinates (0,0) has been changed to (176,144).

[0243] Regarding the men objects 2304 and 2305, coordinate values for the rightward movement have been added to their coordinates. Regarding the objects 2306 and 2307, the coordinates have been changed in correspondence with the change of the coordinates of the background object 2301 from (0,0) to (0,144), so as to move the absolute positions downward by "144". Regarding the object 2308, new coordinates have been generated based on the expansion designation (magnification ratio) and a new distance from the origin (0,0). Regarding the object 2309, its coordinates have been changed in correspondence with the change of the coordinates of the background object 2303 from (0,0) to (176,144), so as to move the absolute position rightward by "176" and downward by "144".

[0244] Note that in horizontal movement of object, the system code synthesizer 2111 merely adds the amount of

movement to the coordinates of display position with respect to the code data of the object, however, in expansion or deformation processing, generates commands corresponding to those processing and newly performs coding. Note that system code in the MPEG4 standard is similar to the CG language VRML, therefore, detailed commands are approximately similar to those in the VRML or ISO/IEC14496-1.

[0245] On the other hand, the header processor 2112 generates a new header in correspondence with the results of editing of the system code data. Fig. 49 is a block diagram showing the detailed construction of the header processor 2112. In Fig. 49, numeral 2120 denotes a separator which separates input header information for respective codes and determine output destinations; 2121, a profile determination unit; 2122, an object number determination unit; 2123, a bit rate determination unit; and 2124, a profile determination unit which determines a profile.

[0246] In the header processor 2112, the separator 2120 extracts PLI code, VOSC and bitrate code from header information of the respective objects, from video code memories 2105 to 2108, and inputs the extracted code into the profile determination unit 2121, the object number determination unit 2122 and the bit rate determination unit 2123. The profile determination unit 2121 decodes the PLI code and detects the highest profile and level from profiles and levels of images to be synthesized. The object number determination unit 2122 counts the number of objects included in the code data by counting the VOSC. The bit rate determination unit 2123 detects the respective bit rates by decoding the bitrate code, and obtains the total sum of the bit rates. The outputs from the respective determination units are inputted into the profile determination unit 2124.

[0247] The profile determination unit 2124 determines profile and level satisfying the highest profile, the number of objects and bit rate, by referring to the profile table as shown in Fig. 28. In the present embodiment, the highest profile of the four images to be synthesized is Core profile and level 2, the number of objects of the synthesized images is 10, and the total sum of the bit rates is 684 kbps. Accordingly, the profile and level satisfying these conditions is, according to the profile table, Main profile and level 3. The profile determination unit 2124 generates new PLI code based on Main profile and level 3, and outputs the PLI code.

[0248] The multiplexer 2114 multiplexes the system code data generated by the system code synthesizer 2111 and the code data of moving image. The moving image code data is reproduced by reading the code, where profile-related code or the like is corrected, from the header processor 2112, or arbitrarily reading the code data stored in the video code memories 2105 to 2108, and multiplexing the read data. Then, the multiplexed code data is outputted to the storage device 2209 and the decoder 2210.

[Processing Procedure]

[0249] Fig. 50E shows code data obtained as a result of multiplexing by the multiplexer 2114. It is understood from Fig. 50E that all the code data shown in Figs. 50A to 50D are synthesized, i.e., all the objects in Figs. 46A to 46D are included. Note that in the multiplexed code data, user data may be positioned prior to the code data of the respective objects, or intensively positioned in a predetermined position within the code data.

[0250] Fig. 51 is a flowchart showing image processing according to the present embodiment. When the apparatus has been started, code data of images are inputted from the respective image input means (storage devices 2201 and 2202, encoder 2204 and communication interface 2206), and stored into the code memories 2101 to 2104 and 2105 to 2108 (step S101). Then, the code data are respectively decoded, and images represented by the decoded data are presented to the user (step S102). Thereafter, the results of the user's edition at the editing operation unit 2207 is obtained (step S103), and the system code is changed (step S104) in accordance with the obtained results of editing. Further, the header of moving image code data are changed in accordance with the profile and level, the number of objects, the bit rate and the like, so as to generate new code (step S105). Then, in the multiplexer 2114, the system code data and video code data are multiplexed and outputted (step S106).

[0251] As code data synthesized by the image editing unit 2208 is inputted into the decoder 2210, the decoder 2210 easily detects the scale of input code data to be decoded, the number of necessary decoders and the like. Accordingly, it can be easily determined whether or not decoding is possible without actually decoding the code data. For example, even if it is determined that decoding is impossible, the code data can be temporarily stored into the storage device 2209 and decoded when a necessary number of decoders are provided.

[0252] Note that the arrangement of the system code memories 2101 to 2104 and the video code memories 2105 to 2108 of the present embodiment is not limited to that in Fig. 47, however, more code memories may be provided, or one memory may be divided into a plurality of areas. Further, a storage medium such as a magnetic disk may be employed.

[0253] According to the present embodiment, when code data of different profiles and/or levels are synthesized, profile and level are re-defined. Since the scale of code data to be inputted, a necessary number of decoders and the like are obtained in advance, in the decoder 2210, it can he easily determined whether or not decoding is possible.

Eleventh Embodiment

[0254] Hereinbelow, an eleventh embodiment of the present invention will he described. Note that the general construction of the moving image processing apparatus of the eleventh embodiment is similar to that of the above-described tenth embodiment in Fig. 45, therefore, an explanation of the construction will be omitted. In the eleventh embodiment, the user designates an arbitrary profile using the editing operation unit 2207, and the image editing unit 2208 generates code data based on the designated profile.

[Construction]

[0255] Fig. 52 is a block diagram showing the detailed construction of the image editing unit 2208 according to the eleventh embodiment. In Fig. 52, elements corresponding to those in Fig. 47 of the tenth embodiment have the same reference numerals, and explanations of the elements will be omitted. In the eleventh embodiment, the MPEG4 coding method is employed as a moving image coding method, however, any other coding method is applicable as long as it encodes a plurality of objects within an image.

[0256] Numeral 2130 denotes a profile controller which performs various controls to synthesize input plural image data in correspondence with a profile designated from the editing operation unit 2207, 2131, a system code synthesizer which synthesises system code; 2132, a header processor which synthesizes and changes header of video code; 2134, a code length regulator which regulates code lengths of respective objects; 2136, an integration processor which performs integration processing on objects; and 2133, 2135 and 2137, selectors which switch respective input/output in accordance with an instruction from the profile controller 2130.

[0257] As in the case of the above-described tenth embodiment, the code data inputted from the storage devices 2201, 2202, the encoder 2204 and the communication interface 2206 are separated into system code data and moving image code data, and stored into the system code memories 2101 to 2104 and the video code memories 2105 to 2108.

[0258] Note that in the eleventh embodiment, the code data inputted from the storage devices 2201 and 2202, the encoder 2204 and the communication interface 2206 are the same as those in the above-described tenth embodiment. Accordingly, the respective images are the same as those in Figs. 46A to 46D, and the code data in Figs. 50A to 50D are obtained by encoding the respective images. Note that in the eleventh embodiment, code data (VO data A) of Core profile and level 2 and at a bit rate of 1024 kbps is inputted from the storage device 2201. Similarly, code data (VO data B) of Core profile and level 1 and at a bit rate of 384 kbps is inputted from the storage device 2202. Similarly, code data (VO data C) of Simple profile and level 3 and at a bit rate of 384 kbps is inputted from the encoder 2204, and code data (VO data D) of Core profile and level 2 and at a bit rate of 768 kbps is inputted from the communication interface 2206.

[0259] In this embodiment, these code data have information unique to the respective objects as user data. The objects in the eleventh embodiment are "people", "background" and "non-cut-out screen image". As user data of a "man" object, information indicating that the type of objects is "man", personal information of the man (sex, age, profession and the like), further, action of the man in the image (e.g., the men objects 2304 and 2305 are discussing, the man object 2307 is giving an injection to the girl object 2306). These object-unique information are utilized upon editing operation such as object search.

[0260] When the respective code data have been stored into the code memories, the video decoder 2109 and the system decoder 2110 respectively decode the code data and output the decoded data to the editing operation unit 2207. At the editing operation unit 2207, the user operates settings such as selection of deletion/holding objects, change of arrangement, moving image start timing and frame rate, thus, the synthesized image 2320 as shown in Fig. 48 is obtained, as in the case of the tenth embodiment.

[Setting of Profile and Level]

[0261] As described above, in the eleventh embodiment, the user can arbitrarily set the profile and level of code data to be outputted, from the editing operation unit 2207. Accordingly, when the generated code data is delivered by broadcasting or the like, the user can adjust the profile and level of the code data to those of a decoder to receive the code data. Hereinbelow, a case where the user has designated Core profile and level 2 at the editing operation unit 2207 will be described.

[0262] The user's designation of profile and level is inputted, with the results of editing, into the profile controller 2130. The synthesized image 2320 shown in Fig. 48 includes 10 objects, and the total sum of the bit rates is 2560 kbps. Further, in Core profile and level 2 designated by the user, the maximum number of objects is 8, and the maximum bit rate is 2048 kbps, according to the profile table in Fig. 28. To perform decoding of the designated profile at the designated level, the number of objects of the synthesized image must be reduced by two, and the bit rate must be controlled to 2048 kbps.

[0263] The profile controller 2130 reduces code length of code data based on the following conditions in the numerical

priority order.

(1) Code length is reduced from the highest profile level
(2) Code length is reduced from a highest bit rate
(3) All the code lengths are reduced

[0264] Hereinbelow, the bit rate of the VO data A is reduced from 1024 kbps to 512 kbps by reducing the code length of the VO data A based on these conditions.

[0265] Further, to reduce the number of objects, two objects may be synthesized into one object, for example. In the eleventh embodiment, objects to he integrated are determined from a plurality of objects, by referring to node information in the system code stored in the system code memories 2101 to 2104. That is, parent-child relation of nodes are referred to, and objects having the same parent are integrated.

[0266] Hereinbelow, the object integration processing according to the eleventh embodiment will be described.

[0267] Figs. 53A to 53D show node statuses of the respective objects in the eleventh embodiment. Fig. 53A shows a node relation of the image in Fig. 46A. The code data is divided into the background 2300 and People node representing people, further, the People node is a parent of the "men" objects 2304 and 2305. Similarly, Fig. 53B shows a node relation of the image in Fig. 46B; Fig. 53C, a node relation of the image in Fig. 46C; and Fig. 53D, a node relation of the image in Fig. 46D. That is, in Fig. 53A, the "men" objects 2304 and 2305 are connected to the People node; in Fig. 53B, the "girl" object 2306 and the "doctor" object 2307 are connected to the People node; in Fig. 53D, the "woman" object 2308 and the "man" object 2309 are connected to a dancer node.

[0268] Accordingly, in the eleventh embodiment, the objects connected to the People and dancer nodes indicative of people are determined as objects to be integrated for respective images. That is, in the image in Fig. 46A, the objects 2304 and 2305 are integrated. Similarly, in the image in Fig. 46B, the objects 2306 and 2307 are integrated; in the image in Fig. 46D, the objects 2308 and 2309 are integrated. By this integration, the number of objects in the synthesized image becomes seven, and the number of objects satisfies Core profile and level 2.

[0269] The profile controller 2130 instructs the system code synthesizer 2131 to newly reproduce the arrangement information of the respective objects after the object integration. The system code synthesizer 2131 generates system code data in the state where the objects are integrated, as in the case of the tenth embodiment.

[0270] At the same time, the profile controller 2130 instructs the header processor 2132 to newly generate header information of the respective objects after the object integration. That is, the size of image is changed to CIF(352×288), the bit rate is set to 2048 kbps, and the PLI code is set to Core profile and level 2. Further, code such as VOL_width, VOL_height, VOP_width, VOP_heigh and bitrate of the integrated objects are corrected.

[0271] The selector 2133 switches a data path so as to pass the object of the image in Fig. 46A (VO data A) through the code length regulator 2134, and not to pass the other objects through the code length regulator 2134, under the control of the profile controller 2130.

[0272] Fig. 54 is a block diagram showing the construction of the coding length regulator 2134. An object decoder 2141 decodes input video code data, and an object encoder 2142 encodes the decoded data using quantization coefficients greater than those in the initial encoding. That is, the bit rate can be reduced by re-encoding the objects of the image in Fig. 46A by rough quantization.

[0273] The selector 2135 switches a data path such that the combinations of the objects 2304 and 2305, the objects 2306 and 2307 and the objects 2308 and 2309 are inputted the integration processor 2136, under the control of the profile controller 2130.

[0274] The detailed construction of the integration processor 2136 is the same as that of the object integrator 1123 of the seventh embodiment in Fig. 34. Accordingly, explanations of the construction and processing of the integration processor 2136 will be omitted.

[0275] The code data on synthesized color image information and code data on mask information are inputted via a selector 2137 into the multiplexer 2114, and are multiplexed to code data of one object. The result of the system code synthesizer 2131, the header generated by the header processor 2132, and code data corresponding to the header are sequentially inputted via the selector 2137 into the multiplexer 2114, and multiplexed and outputted.

[0276] Fig. 57 shows the data structure of code data outputted from the image editing unit 2208 of the eleventh embodiment. In Fig. 57, in the video object data, the newly-set PLI code (PLIN-1 in Fig. 57) is provided at the head. Then the VO data A-1-1 corresponding to the background object 2300, and the VO data A-1-23 corresponding to the object synthesized from the objects 2302 and 2304 follow. Further, the VO data B-1-1 corresponding to the background object 2301, the VO data B-1-23 corresponding to the object synthesized from the objects 2306 and 2307, the VO data C-1-1 corresponding to the object 2302, the VO data D-1-1 corresponding to the background object 2303, and the VO data D-1-23 corresponding to the object synthesized from the objects 2308 and 2309 follow. That is, seven video objects exist in one Visual Object.

[0277] The code data obtained as above is stored into the storage device 2209, or decoded by the decoder 2210 and

displayed as an image as shown in Fig. 48 on the display unit 2211.

[0278] As described above, according to the eleventh embodiment, when code data of different profiles and levels are synthesized, profile and level are re-defined, and further, the number of objects and the bit rate can be regulated. Thus, code data of profile and level desired by the user can be obtained.

[0279] Further, respective objects within an image can be arbitrarily synthesized by integrating objects based on the relation among the objects (nodes) described in the system code. That is, a synthesizing procedure closer to a user's intuitive synthesizing procedure can be realized.

Modification of Eleventh Embodiment

[0280] Fig. 55 is a block diagram showing a modified construction of the code length regulator 2134 according to the eleventh embodiment. If input video code data has been motion compensated, a Huffman decoder 2143 decodes the quantization DCT coefficients. The Huffman decoder 2143 inputs the obtained quantization DCT coefficients into a high frequency eliminator 2144, to eliminate high frequency components by replacing the high frequency components with "0". Then, a Huffman encoder 2143 encodes the output from the high frequency eliminator 2144. That is, the code length can be reduced by eliminating high frequency components of the object and re-encoding the data.

[0281] Fig. 56 is a block diagram showing another modified construction of the code length regulator 2134. If input video code data has been motion compensated, the Huffman decoder 2143 decodes the quantization DCT coefficients. Then, an inverse quantizer 2146 performs inverse quantization on the obtained quantization DCT coefficients, then a quantizer 2147 quantizes the obtained DCT coefficients using quantization coefficients greater than those used in the initial coding. Then the Huffman encoder 2145 encodes the data. That is, the code length can be reduced by decoding code data of a motion compensated object and re-encoding the data with rough quantization.

[0282] Note that in the eleventh embodiment, objects to be integrated may be selected using information unique to the respective objects, described independently of the user data or code data, in addition to the relation among the objects indicated by nodes. That is, objects having similar attributes ("people", "profession" and the like) may be integrated. Further, the objects 2307 and 2306 may be integrated based on the attributes indicating actions of "people" objects such as "giving an injection" and "taking an injection" as selection conditions.

[0283] Further, objects to be integrated may be selected by the combination of plural conditions such as object size, code length, location relation and user's instruction.

[0284] Further, in the eleventh embodiment, objects are integrated based on the relation among the objects (nodes) described in the system code, however, the number of object may be reduced by deleting objects selected based on the nodes. In this case, the bit rate can be reduced at the same time.

[0285] Note that the arrangement of the system code memories 2101 to 2104 and the video code memories 2105 to 2108 is not limited to that in Fig. 47, but more code memories may be provided or one memory may be divided into a plurality of areas. Further, a storage medium such as a magnetic disk may be employed.

[0286] As described above, according to the tenth and eleventh embodiments, one code data based on a predetermined standard can he obtained by synthesizing a plurality of code data, encoded for a plurality of image information (objects). Further, the synthesized code data may de decoded by a decoder of arbitrary coding specifications. Further, the number of objects and the code length of the code data can be regulated.

[0287] Further, in the above-described respective embodiments, the object 0 is background, however, the object 0 is not limited to the background but may be a moving image of a general object or the like.

[0288] The present invention can be applied to a system constituted by a plurality of devices (e.g., host computer, interface, reader, printer) or to an apparatus comprising a single device (e.g., copy machine, facsimile).

[0289] Further, the object of the present invention can be also achieved by providing a storage medium storing program codes for performing the aforesaid processes to a system or an apparatus, reading the program codes with a computer (e.g., CPU, MPU) of the system or apparatus from the storage medium, then executing the program.

[0290] In this case, the program codes read from the storage medium realize the functions according to the embodiments, and the storage medium storing the program codes constitutes the invention.

[0291] Further, the storage medium, such as a floppy disk, a hard disk, an optical disk, a magneto-optical disk, CD-ROM, CD-R, a magnetic tape, a non-volatile type memory card, and ROM can be used for providing the program codes.

[0292] Furthermore, besides aforesaid functions according to the above embodiments are realized by executing the program codes which are read by a computer, the present invention includes a case where an OS (operating system) or the like working on the computer performs a part or entire processes in accordance with designations of the program codes and realizes functions according to the above embodiments.

[0293] Furthermore, the present invention also includes a case where, after the program codes read from the storage medium are written in a function expansion card which is inserted into the computer or in a memory provided in a function expansion unit which is connected to the computer, CPU or the like contained in the function expansion card or unit performs a part or entire process in accordance with designations of the program codes and realizes functions of the

above embodiments.

[0294] The present invention is not limited to the above embodiments and various changes and modifications can be made within the spirit and scope of the present invention. Therefore, to appraise the public of the scope of the present invention, the following claims are made.

Claims

1. A data processing apparatus having decoding means for decoding code encoded in image object units, said apparatus comprising:

   detection means for detecting a number of objects included in input code and a number of objects decodable by said decoding means; and
   control means for controlling the number of objects of the input code, based on the number of objects and the number of decodable objects detected by said detection means.

2. The apparatus according to claim 1, wherein if said number of objects is greater than said number of decodable objects, said control means reduces the number of objects included in said code to said number of decodable objects.

3. The apparatus according to claim 1, further comprising:

   extraction means for extracting location information of the objects included in said code; and
   combining means for combining code of a plurality of objects, based on an instruction from said control means and the location information extracted by said extraction means.

4. The apparatus according to claim 3, wherein said combining means combines code of a plurality of objects away from each other by a distance therebetween, wherein said distance being shorter than other distances between objects calculated from said location information.

5. The apparatus according to claim 1, further comprising:

   extraction means for extracting motion information indicative of motions of the objects included in said code; and
   combining means for combining a plurality of objects based on an instruction from said control means and the motion information extracted by said extraction means.

6. The apparatus according to claim 5, wherein said combining means combines code of a plurality of objects having the motion information similar to each other.

7. The apparatus according to claim 1, further comprising:

   extraction means for extracting code lengths of the objects included in said code; and
   combining means for combining a plurality of objects based on an instruction from said control means and the code lengths extracted by said extraction means.

8. The apparatus according to claim 7, wherein said combining means combines code of a plurality of objects having code lengths shorter than other code lengths.

9. The apparatus according to claim 1, further comprising initialization means for determining a coding method for encoding the input code in frame units, and initializing said control means based on the result of determination.

10. The apparatus according to claim 9, wherein said initialization means determines whether said code is encoded based on interframe correlation or encoded based on intraframe information.

11. The apparatus according to claim 10, wherein if said code is encoded based on the intraframe information, said initialization means initializes said control means.

12. The apparatus according to claim 1, wherein said code is code of a still image.

13. The apparatus according to claim 1, wherein said code is code of a moving image.

14. A data processing method for decoding code encoded in image object units, said method comprising the steps of:

detecting the number of objects included in input code and the number of objects decodable by said means; and
controlling the number of objects of the input code, based on the number of objects and the number of decodable objects detected at said detection step.

15. The method according to claim 14, wherein at said control step, if said number of objects is greater than said number of decodable objects, the number of objects included in said code is reduced to said number of decodable objects.

16. A computer program product comprising a computer readable medium having computer program code, for executing data processing which decodes code encoded in image object units, said product comprising:

detecting procedure code for detecting a number of objects included in input code and a number of decodable objects; and
controlling procedure code for controlling the number of objects of the input code, based on the number of objects and the number of decodable objects detected in said detection procedure.

17. A data processing apparatus for processing a data array to reproduce an image with a plurality of coded image objects, said apparatus comprising:

detection means for detecting a number of image objects included in said data array; and
control means for controlling the number of image objects included in said data array based on the number of image objects detected by said detection means.

18. The apparatus according to claim 17, wherein

if said number of image objects is greater than a predetermined number, said control means reduces the number of image objects included in said data array.

19. The apparatus according to claim 18, wherein said predetermined number is a number of objects which can be processed by decoding means for decoding said data array.

20. The apparatus according to claim 18, wherein said control means reduces the number of image objects by deleting an image object.

21. The apparatus according to claim 20, wherein said control means obtains code lengths of the respective image objects in said data array, and deletes the image object based on the obtained code lengths.

22. The apparatus according to claim 21, wherein said control means deletes sequentially from an image object having the shortest code length.

23. The apparatus according to claim 20, wherein said control means obtains image sizes of the respective image objects in said data array, and deletes the image object based on the obtained image sizes.

24. The apparatus according to claim 23, wherein said control means deletes sequentially from an image object having the minimum size.

25. The apparatus according to claim 20, further comprising setting means for setting a priority order of the image objects in said data array,
wherein said control means deletes the image object based on the priority order set by said setting means.

26. The apparatus according to claim 20, wherein said control means reduces the number of the image objects by integrating a plurality of image objects.

27. The apparatus according to claim 26, further comprising selection means for selecting a plurality of image objects included in said data array,

wherein said control means integrates the plurality of image objects selected by said selection means.

28. The apparatus according to claim 27, wherein said control means comprises decoding means for decoding the plurality of image objects selected by said selection means, synthesizing means for synthesizing the plurality of image objects decoded by said decoding means, and coding means for encoding an image object synthesized by said synthesizing means, so as to integrate the plurality of image objects selected by said selection means.

29. The apparatus according to claim 28, wherein said control means further comprises counting means for counting code lengths of the plurality of image objects selected by said selection means,

and wherein said coding means controls coding parameters based on the results of counting by said counting means.

30. The apparatus according to claim 27, wherein said control means comprises separation means for separating the plurality of image objects selected by said selection means into color information and mask information, color information synthesizing means for synthesizing the color information separated by said separation means, mask information synthesizing means for synthesizing the mask information separated by said separation means, and multiplexing means for multiplexing the color information synthesized by said color information synthesizing means and the mask information synthesized by said mask information synthesizing means.

31. The apparatus according to claim 27, wherein said selection means enables manual selection of a plurality of image objects.

32. The apparatus according to claim 27, wherein said selection means selects a plurality of image objects based on spatial location information of the respective image objects.

33. The apparatus according to claim 27, wherein said selection means selects a plurality of image objects, one of which including the other.

34. The apparatus according to claim 27, wherein said selection means selects a plurality of image objects away from each other by a distance less than a predetermined value.

35. The apparatus according to claim 27, wherein said selection means obtains code lengths of the respective image objects in said data array, and selects an image object based on the obtained code lengths.

36. The apparatus according to claim 35, wherein said selection means selects sequentially from an image object having the shortest code length.

37. The apparatus according to claim 27, wherein said selection means obtains image sizes of the respective image objects in said data array, and selects the image object based on the obtained image sizes.

38. The apparatus according to claim 37, wherein said selection means selects sequentially from an image object having the minimum image size.

39. The apparatus according to claim 37, further comprising setting means for setting a priority order of the image objects in said data array,

wherein said selection means selects the image object based on the priority order set by said setting means.

40. The apparatus according to claim 17, wherein said data array is code data adapted to or based on the MPEG4 standard.

41. A data processing method for processing a data array to reproduce an image with a plurality of coded image objects, said method comprising the steps of:

detecting a number of image objects included in said data array; and
controlling the number of image objects included in said data array based on the number of image objects detected at said detection step.

42. A computer program product comprising a computer readable medium having computer program code, for executing data processing which processes a data array to reproduce an image with a plurality of coded image objects, said product comprising:

detection procedure code for detecting a number of image objects included in said data array; and
control procedure code for controlling the number of image objects included in said data array based on the number of image objects detected in said detection procedure.

43. A data processing apparatus comprising:

input means for inputting a plurality of image data to construct one frame, wherein said image data respectively including N image objects, where N≥1 holds; and
generation means for generating image data having M image objects, where M≥1 holds, constructing said one frame, by integrating at least a part of said N image objects based on additional information indicative of relation among the image objects.

44. The apparatus according to claim 43, wherein said M image objects are an appropriate number of image objects to be processed in accordance with a predetermined coding standard.

45. The apparatus according to claim 44, wherein if the number of image objects included in said image data inputted by said input means is greater than the number of objects defined by said predetermined coding standard, said generation means performs integration processing.

46. The apparatus according to claim 43, wherein said image data inputted by said input means is adapted to or based on the MPEG4 standard.

47. A data processing method comprising the steps of:

inputting a plurality of image data to construct one frame, wherein said image data respectively including N image objects, where N≥1 holds; and
generating image data having M image objects, where M≥1 holds, constructing said one frame, by integrating at least a part of said N image objects based on additional information indicative of relation among the image objects.

48. A computer program product comprising a computer readable medium having computer program code, for executing data processing, said product comprising:

input procedure code for inputting a plurality of image data to construct one frame, wherein said image data respectively including N image objects, where N≥1 holds; and
generation procedure code for generating image data having M image objects, where M≥1 holds, constructing said one frame, by integrating at least a part of said N image objects based on additional information indicative of relation among the image objects.

49. A data processing apparatus for processing a data array to reproduce one frame image with a plurality of coded image objects, said apparatus comprising:

input means for inputting a plurality of data arrays;
instruction means for instructing synthesizing of a plurality of data arrays inputted by said input means;
designation means for designating coding specifications of a processed data array;
control means for controlling information amounts of the plurality of data arrays inputted by said input means, based on the coding specifications designated by said designation means; and
synthesizing means for synthesizing the plurality of data arrays with information amounts controlled by said control means, based on the coding specifications designated by said designation means.

50. The apparatus according to claim 49, wherein said synthesizing means sets coding specifications of synthesized data array to the coding specifications designated by said designation means.

51. The apparatus according to claim 49, wherein said instruction means instructs synthesizing including change of

spatial locations of the image objects included in said data array.

52. The apparatus according to claim 49, wherein if the number of image objects included in the data array synthesized by said synthesizing means is greater than the number of image objects according to the coding specifications designated by said designation means, said control means reduces the number of image objects included in the data array.

53. The apparatus according to claim 52, wherein said control means reduces the number of image objects by integrating image objects included in the data array.

54. The apparatus according to claim 53, wherein said control means selects image objects to be integrated, based on relation among the image objects included in said data array.

55. The apparatus according to claim 54, wherein the relation among the image objects is represented by node information.

56. The apparatus according to claim 52, wherein said control means reduces the number of image objects by deleting at least one image object included in said data array.

57. The apparatus according to claim 49, wherein if a code length of an image object included in the data array synthesized by said synthesizing means is longer than a value based on the coding specifications designated by said designation means, said control means reduces the code length of the image object.

58. The apparatus according to claim 49, wherein said control means reduces a code length, sequentially from an image object with the highest coding specifications, among the plurality of data arrays to be synthesized, included in a data array.

59. The apparatus according to claim 49, wherein said control means reduces a code length of an image object by performing re-encoding with rough quantization coefficients, on a data array.

60. The apparatus according to claim 49, wherein said control means reduces a code length of an image object by performing re-encoding which eliminates high frequency components, on a data array.

61. The apparatus according to claim 49, further comprising transmission means for transmitting the data array synthesized by said synthesizing means.

62. The apparatus according to claim 49, further comprising decoding means for decoding the data array synthesized by said synthesizing means.

63. The apparatus according to claim 58, further comprising display means for displaying an image represented by the data array decoded by said decoding means.

64. The apparatus according to claim 49, wherein said coding specifications are adapted to or based on the MPEG4 standard.

65. The apparatus according to claim 49, wherein said data array is adapted to or based on the MPEG4 standard.

66. A data processing method for processing a data array to reproduce one frame image with a plurality of coded image objects, said method comprising the steps of:

inputting a plurality of data arrays;
instructing synthesizing of a plurality of data arrays inputted at said input step;
designating coding specifications of a processed data array;
controlling information amounts of the plurality of data arrays inputted at said input step, based on the coding specifications designated at said designation step; and
synthesizing the plurality of data arrays with information amounts controlled at said control step, based on the coding specifications designated at said designation step.

67. A computer program product comprising a computer readable medium having computer program code, for executing data processing which processes a data array to reproduce one frame image with a plurality of coded image objects, said product comprising;

input procedure code for inputting a plurality of data arrays;
instruction procedure code for instructing synthesizing of a plurality of data arrays inputted in said input procedure;
designation procedure code for designating coding specifications of a processed data array;
control procedure code for controlling information amounts of the plurality of data arrays inputted in said input procedure, based on the coding specifications designated in said designation procedure; and
synthesizing procedure code for synthesizing the plurality of data arrays with information amounts controlled in said control procedure, based on the coding specifications designated in said designation procedure.

FIG. 1

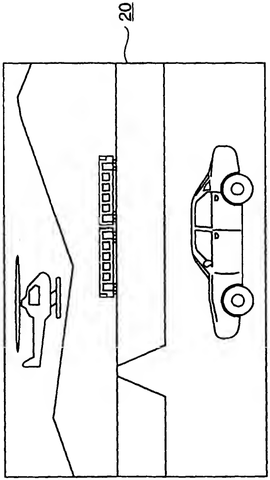


FIG. 2

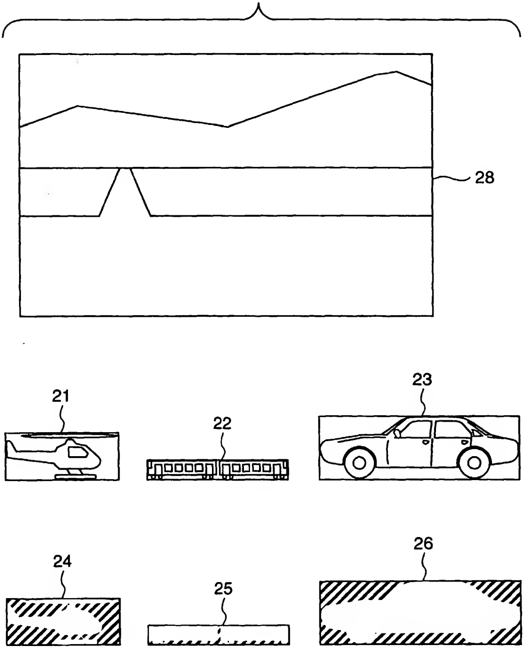
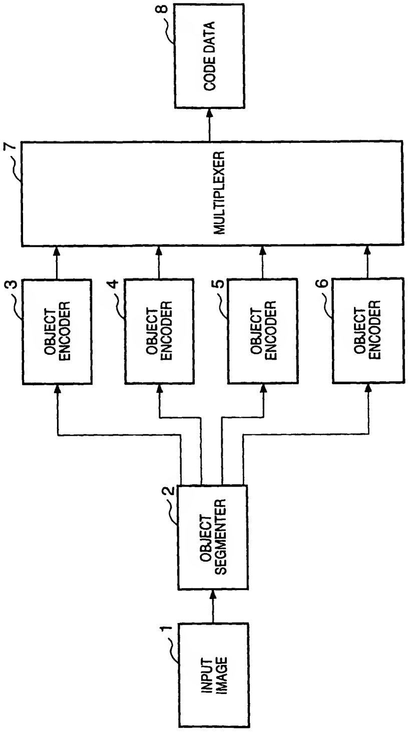


FIG. 3



INPUT IMAGE
1
OBJECT SEGMENTER
2
OBJECT ENCODER
3
OBJECT ENCODER
4
OBJECT ENCODER
5
OBJECT ENCODER
6
MULTIPLEXER
7
CODE DATA
8

FIG. 4

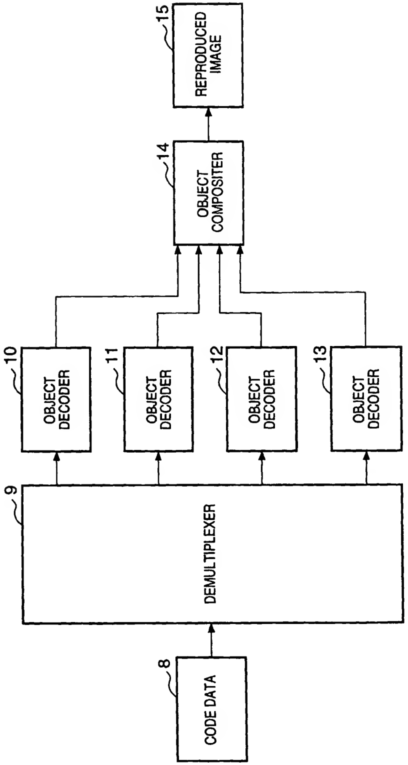

15
REPRODUCED IMAGE
14
OBJECT COMPOSITER
10
OBJECT DECODER
11
OBJECT DECODER
12
OBJECT DECODER
13
OBJECT DECODER
9
DEMULTIPLEXER
8
CODE DATA

FIG. 5

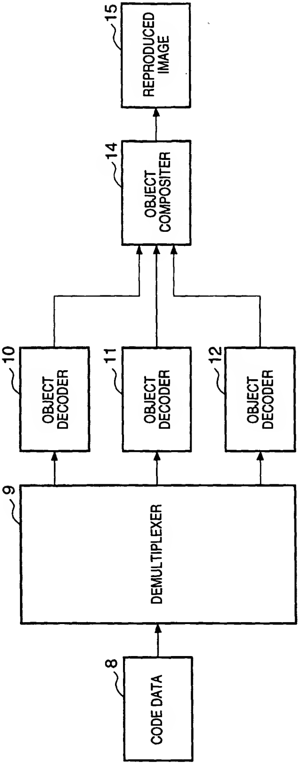

CODE DATA
8
DEMULTIPLEXER
9
OBJECT DECODER
10
OBJECT DECODER
11
OBJECT DECODER
12
OBJECT COMPOSITER
14
REPRODUCED IMAGE
15

FIG. 6

38

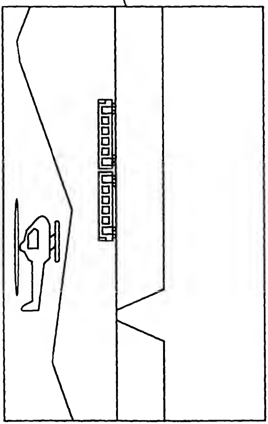


FIG. 7

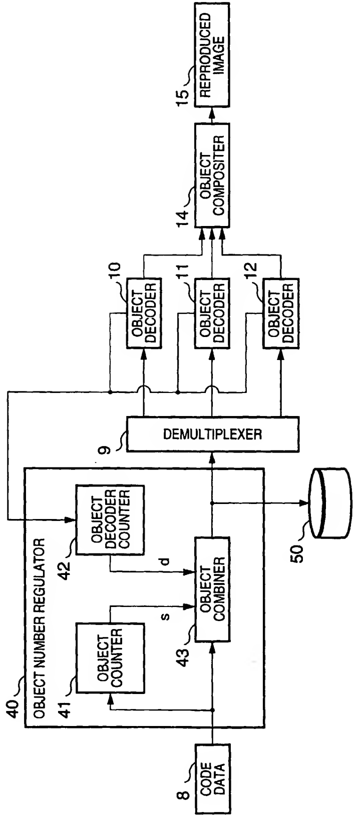

8
CODE DATA
40
OBJECT NUMBER REGULATOR
41
OBJECT COUNTER
42
OBJECT DECODER COUNTER
43
OBJECT COMBINER
s
d
50
9
DEMULTIPLEXER
10
OBJECT DECODER
11
OBJECT DECODER
12
OBJECT DECODER
14
OBJECT COMPOSITER
15
REPRODUCED IMAGE

FIG. 8

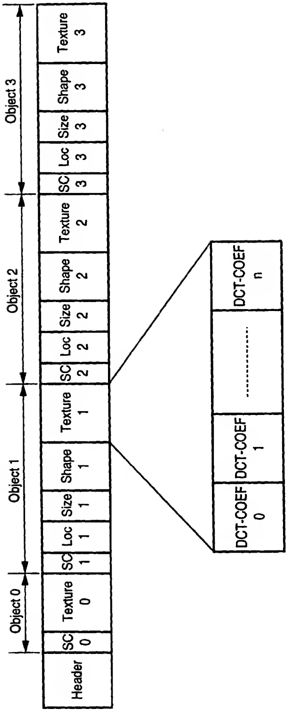

Header
SC 0
Texture 0
Object 0
SC 1
Loc 1
Size 1
Shape 1
Texture 1
Object 1
SC 2
Loc 2
Size 2
Shape 2
Texture 2
Object 2
SC 3
Loc 3
Size 3
Shape 3
Texture 3
Object 3
DCT-COEF 0
DCT-COEF 1
DCT-COEF n

FIG. 9

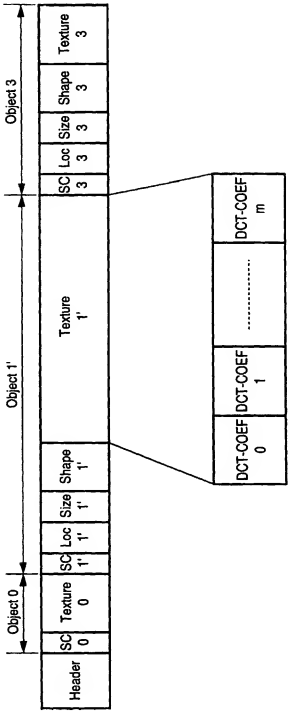

Header
SC 0
Texture 0
Object 0
SC 1'
Loc 1'
Size 1'
Shape 1'
Texture 1'
Object 1'
SC 3
Loc 3
Size 3
Shape 3
Texture 3
Object 3
DCT-COEF 0
DCT-COEF 1
DCT-COEF m

FIG. 10

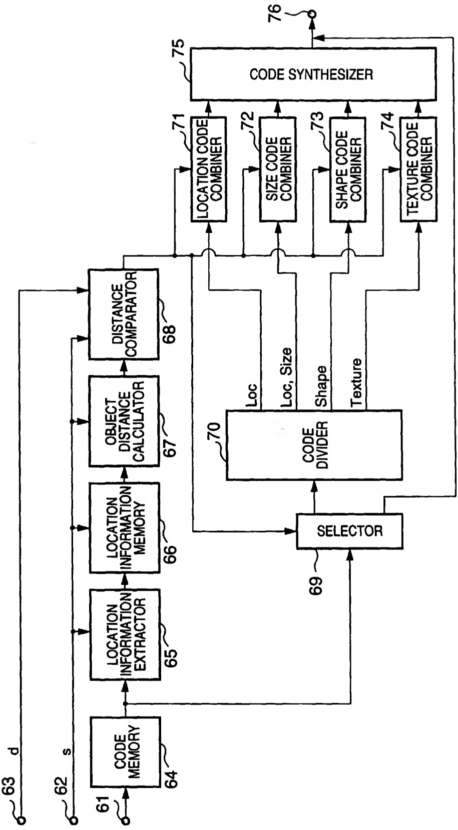

61
62
63
s
d
CODE MEMORY
64
LOCATION INFORMATION EXTRACTOR
65
LOCATION INFORMATION MEMORY
66
OBJECT DISTANCE CALCULATOR
67
DISTANCE COMPARATOR
68
SELECTOR
69
CODE DIVIDER
70
Loc
Loc, Size
Shape
Texture
LOCATION CODE COMBINER
71
SIZE CODE COMBINER
72
SHAPE CODE COMBINER
73
TEXTURE CODE COMBINER
74
CODE SYNTHESIZER
75
76

FIG. 11A

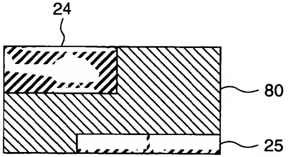


FIG. 11B

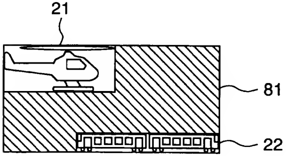


FIG. 12

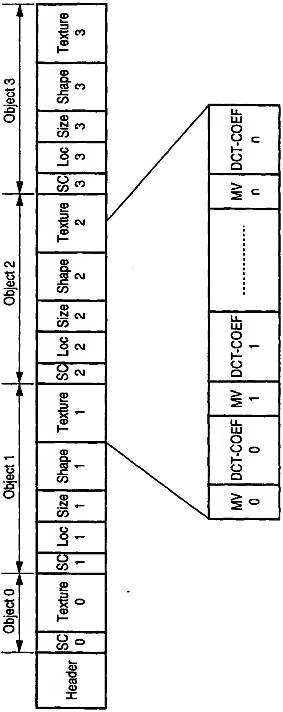

Header
SC 0
Texture 0
Object 0
SC 1
Loc 1
Size 1
Shape 1
Texture 1
Object 1
SC 2
Loc 2
Size 2
Shape 2
Texture 2
Object 2
SC 3
Loc 3
Size 3
Shape 3
Texture 3
Object 3
MV 0
DCT-COEF 0
MV 1
DCT-COEF 1
MV n
DCT-COEF n

FIG. 13

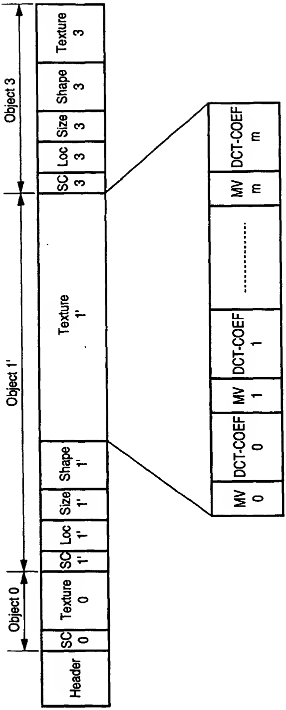

Header
SC 0
Texture 0
Object 0
SC 1'
Loc 1'
Size 1'
Shape 1'
Texture 1'
Object 1'
SC 3
Loc 3
Size 3
Shape 3
Texture 3
Object 3
MV 0
DCT-COEF 0
MV 1
DCT-COEF 1
MV m
DCT-COEF m

FIG. 14A

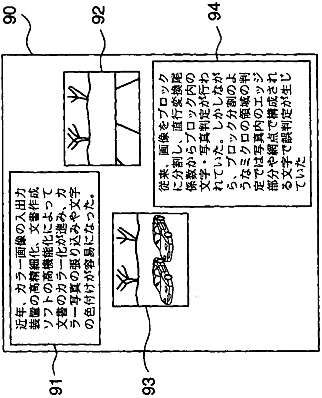


FIG. 14B

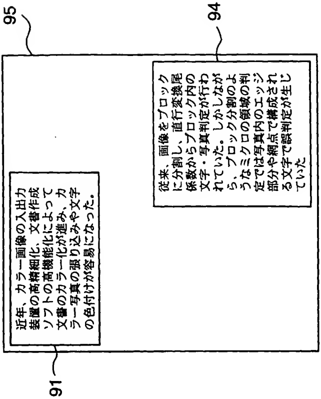


FIG. 14C

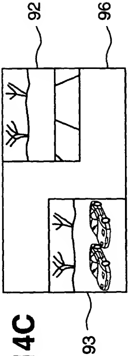


FIG. 15

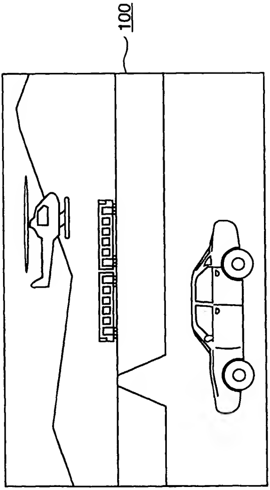


FIG. 16

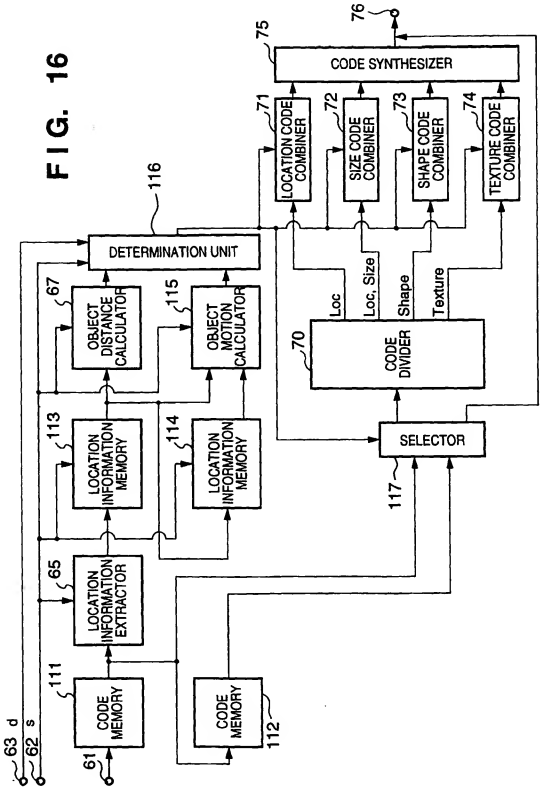

CODE SYNTHESIZER
LOCATION CODE COMBINER
SIZE CODE COMBINER
SHAPE CODE COMBINER
TEXTURE CODE COMBINER
DETERMINATION UNIT
OBJECT DISTANCE CALCULATOR
OBJECT MOTION CALCULATOR
LOCATION INFORMATION MEMORY
LOCATION INFORMATION MEMORY
LOCATION INFORMATION EXTRACTOR
CODE MEMORY
CODE MEMORY
CODE DIVIDER
SELECTOR
Loc
Loc, Size
Shape
Texture
63 d
62 s
61
111
112
65
113
114
67
115
116
117
70
71
72
73
74
75
76

FIG. 17A

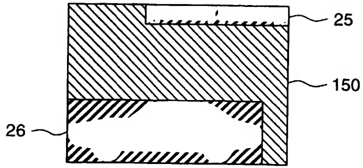


FIG. 17B

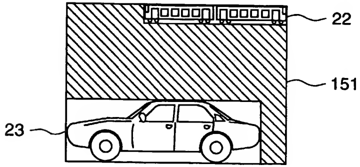


FIG. 18

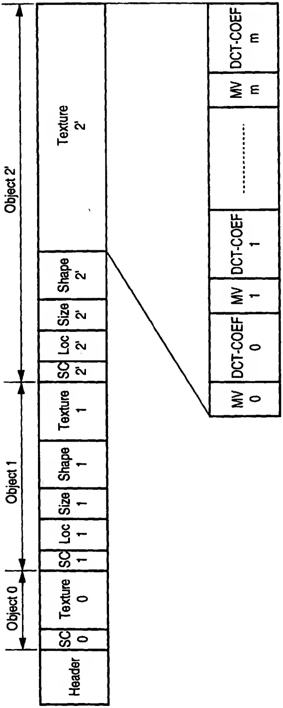

Object 0
Object 1
Object 2'
Header
SC 0
Texture 0
SC 1
Loc 1
Size 1
Shape 1
Texture 1
SC 2'
Loc 2'
Size 2'
Shape 2'
Texture 2'
MV 0
DCT-COEF 0
MV 1
DCT-COEF 1
MV m
DCT-COEF m

FIG. 19

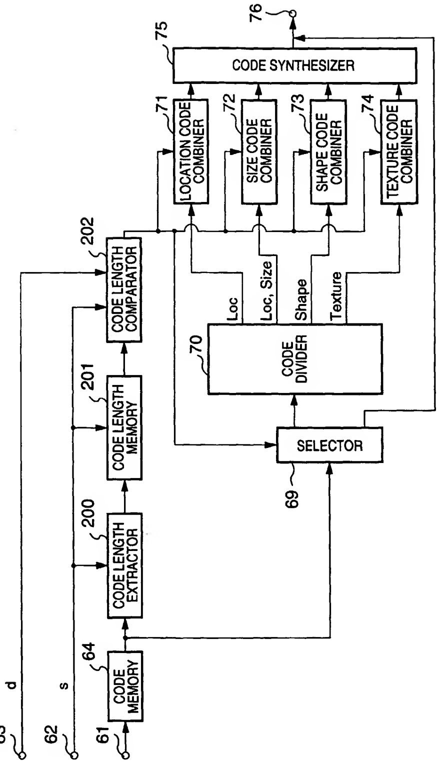
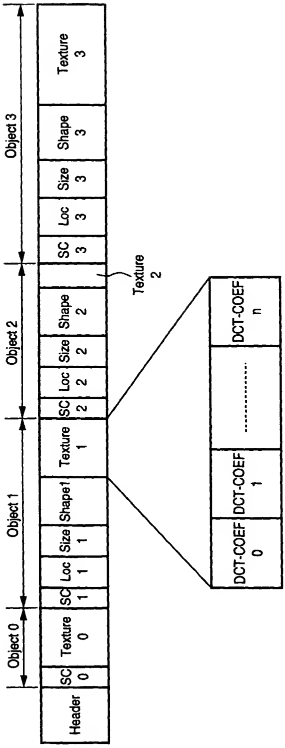

63
62
61
d
s
CODE MEMORY
64
CODE LENGTH EXTRACTOR
200
CODE LENGTH MEMORY
201
CODE LENGTH COMPARATOR
202
SELECTOR
69
CODE DIVIDER
70
Loc
Loc, Size
Shape
Texture
LOCATION CODE COMBINER
71
SIZE CODE COMBINER
72
SHAPE CODE COMBINER
73
TEXTURE CODE COMBINER
74
CODE SYNTHESIZER
75
76

FIG. 20


Header
SC 0
Texture 0
Object 0
SC 1
Loc 1
Size 1
Shape1
Texture 1
Object 1
SC 2
Loc 2
Size 2
Shape 2
Texture 2
Object 2
SC 3
Loc 3
Size 3
Shape 3
Texture 3
Object 3
DCT-COEF 0
DCT-COEF 1
DCT-COEF n

F I G. 21

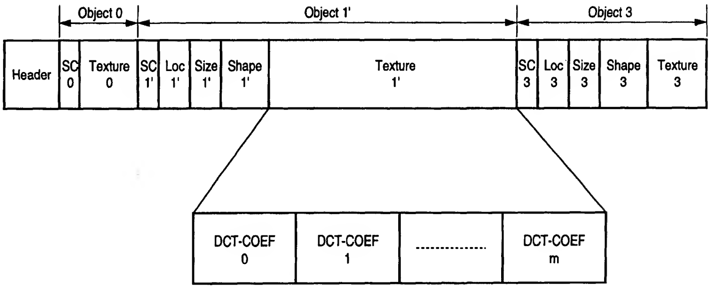

Object 0
Object 1'
Object 3
Header
SC 0
Texture 0
SC 1'
Loc 1'
Size 1'
Shape 1'
Texture 1'
SC 3
Loc 3
Size 3
Shape 3
Texture 3
DCT-COEF 0
DCT-COEF 1
DCT-COEF m

FIG. 22

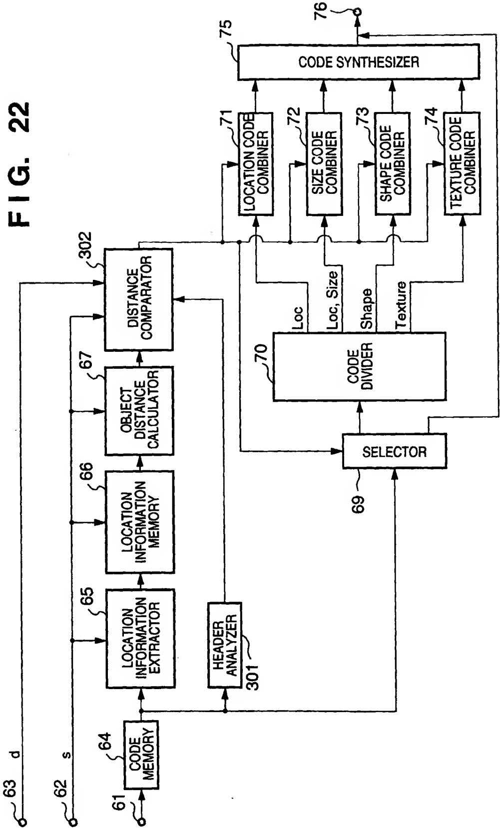

61
62
63
s
d
64
CODE MEMORY
65
LOCATION INFORMATION EXTRACTOR
66
LOCATION INFORMATION MEMORY
67
OBJECT DISTANCE CALCULATOR
302
DISTANCE COMPARATOR
301
HEADER ANALYZER
69
SELECTOR
70
CODE DIVIDER
Loc
Loc, Size
Shape
Texture
71
LOCATION CODE COMBINER
72
SIZE CODE COMBINER
73
SHAPE CODE COMBINER
74
TEXTURE CODE COMBINER
75
CODE SYNTHESIZER
76

FIG. 23

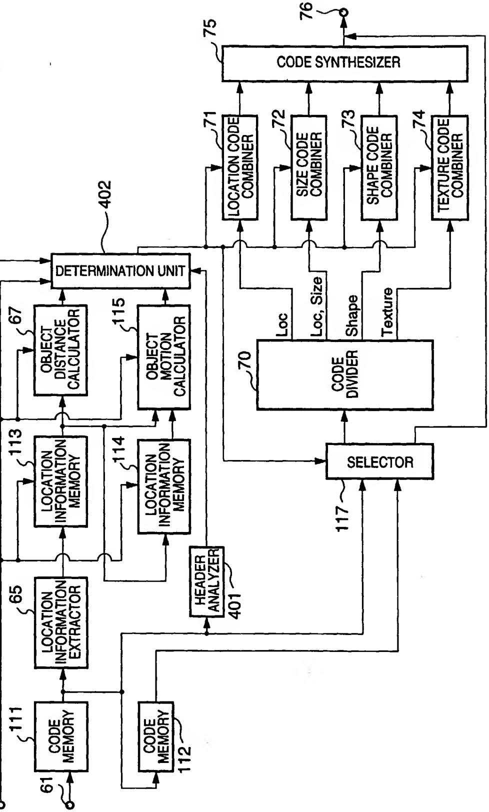

CODE SYNTHESIZER
75
76
LOCATION CODE COMBINER
71
SIZE CODE COMBINER
72
SHAPE CODE COMBINER
73
TEXTURE CODE COMBINER
74
402
DETERMINATION UNIT
Loc
Loc, Size
Shape
Texture
CODE DIVIDER
70
OBJECT DISTANCE CALCULATOR
67
OBJECT MOTION CALCULATOR
115
LOCATION INFORMATION MEMORY
113
LOCATION INFORMATION MEMORY
114
SELECTOR
117
HEADER ANALYZER
401
LOCATION INFORMATION EXTRACTOR
65
CODE MEMORY
111
CODE MEMORY
112
63
d
62
s
61

FIG. 24

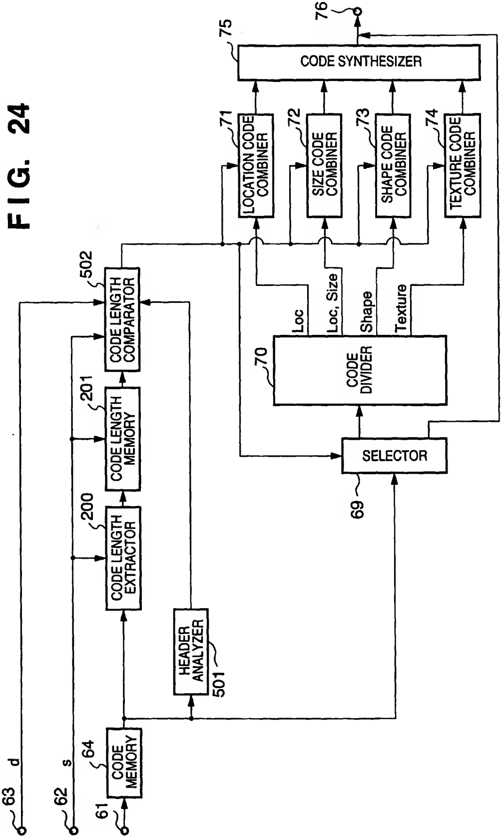

61
62
63
s
d
CODE MEMORY
64
CODE LENGTH EXTRACTOR
200
CODE LENGTH MEMORY
201
CODE LENGTH COMPARATOR
502
HEADER ANALYZER
501
SELECTOR
69
CODE DIVIDER
70
Loc
Loc, Size
Shape
Texture
LOCATION CODE COMBINER
71
SIZE CODE COMBINER
72
SHAPE CODE COMBINER
73
TEXTURE CODE COMBINER
74
CODE SYNTHESIZER
75
76

FIG. 25

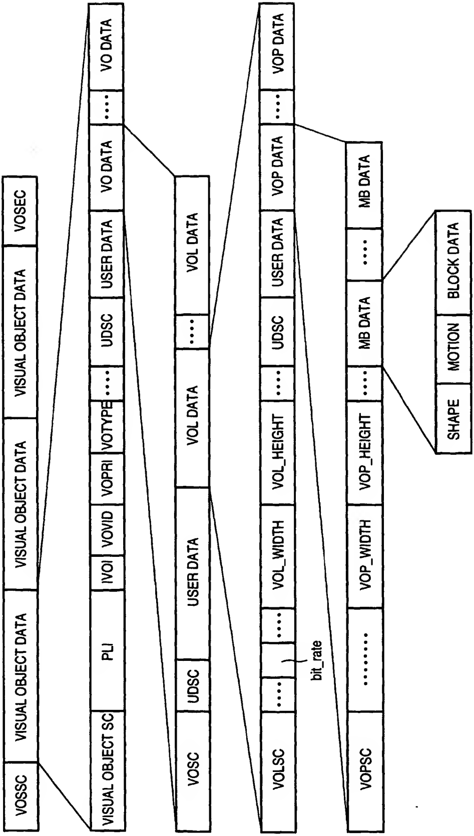

VOSSC
VISUAL OBJECT DATA
VISUAL OBJECT DATA
VISUAL OBJECT DATA
VOSEC
VISUAL OBJECT SC
PLI
IVOI
VOVID
VOPRI
VOTYPE
....
UDSC
USER DATA
VO DATA
....
VO DATA
VOSC
UDSC
USER DATA
VOL DATA
....
VOL DATA
VOLSC
....
bit_rate
....
VOL_WIDTH
VOL_HEIGHT
....
UDSC
USER DATA
VOP DATA
....
VOP DATA
VOPSC
....
VOP_WIDTH
VOP_HEIGHT
....
MB DATA
....
MB DATA
SHAPE
MOTION
BLOCK DATA

FIG. 26

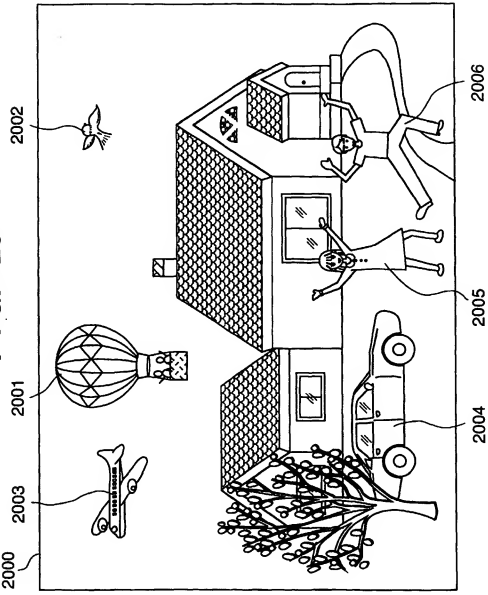


FIG. 27

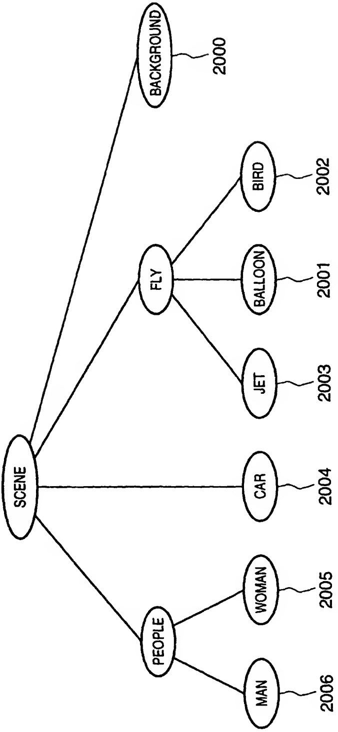

BACKGROUND
2000
FLY
BIRD
2002
BALLOON
2001
JET
2003
SCENE
CAR
2004
PEOPLE
WOMAN
2005
MAN
2006

FIG. 28

PROFILE	LEVEL	MAXIMUM NUMBER OF OBJECTS	MAXIMUM BIT RATE (KBPS)
SIMPLE	1	4	64
	2	4	128
	3	4	384
CORE	1	4	384
	2	8	2000
MAIN	2	8	2000
	3	16	15000

FIG. 29

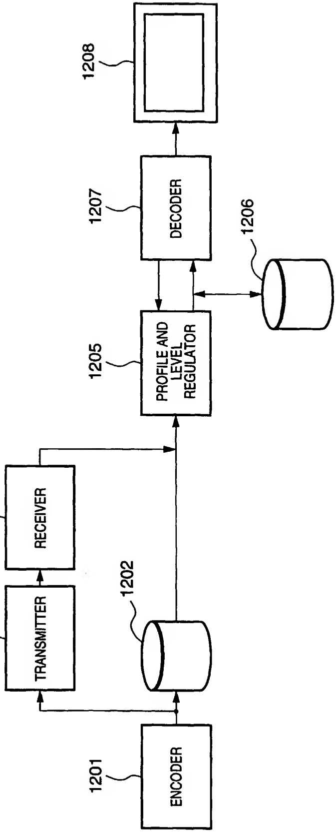

1201
ENCODER
1202
1203
TRANSMITTER
1204
RECEIVER
1205
PROFILE AND LEVEL REGULATOR
1206
1207
DECODER
1208

FIG. 30

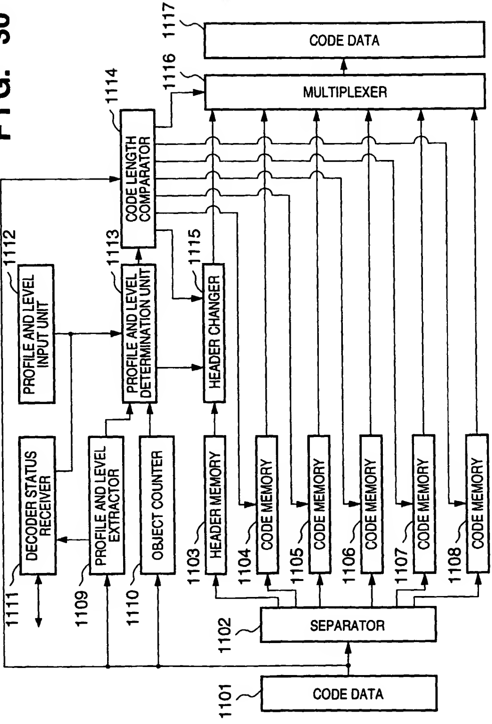

CODE DATA
MULTIPLEXER
CODE LENGTH COMPARATOR
PROFILE AND LEVEL INPUT UNIT
PROFILE AND LEVEL DETERMINATION UNIT
HEADER CHANGER
DECODER STATUS RECEIVER
PROFILE AND LEVEL EXTRACTOR
OBJECT COUNTER
HEADER MEMORY
CODE MEMORY
CODE MEMORY
CODE MEMORY
CODE MEMORY
CODE MEMORY
SEPARATOR
CODE DATA
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117

FIG. 31A

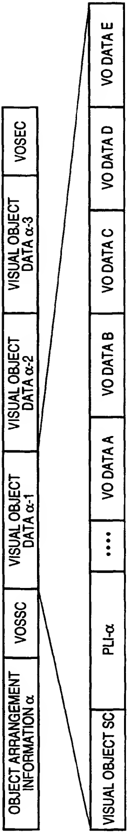


FIG. 31B

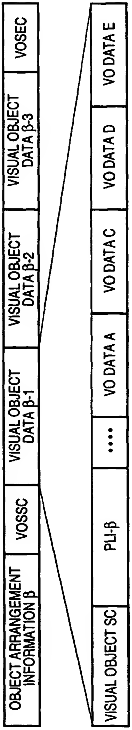


FIG. 32

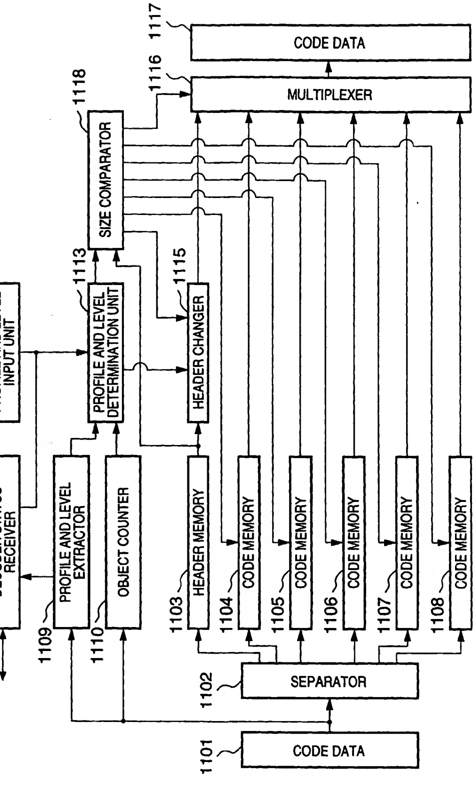

CODE DATA
MULTIPLEXER
SIZE COMPARATOR
PROFILE AND LEVEL INPUT UNIT
PROFILE AND LEVEL DETERMINATION UNIT
HEADER CHANGER
DECODER STATUS RECEIVER
PROFILE AND LEVEL EXTRACTOR
OBJECT COUNTER
HEADER MEMORY
CODE MEMORY
CODE MEMORY
CODE MEMORY
CODE MEMORY
CODE MEMORY
SEPARATOR
CODE DATA
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1115
1116
1117
1118

FIG. 33

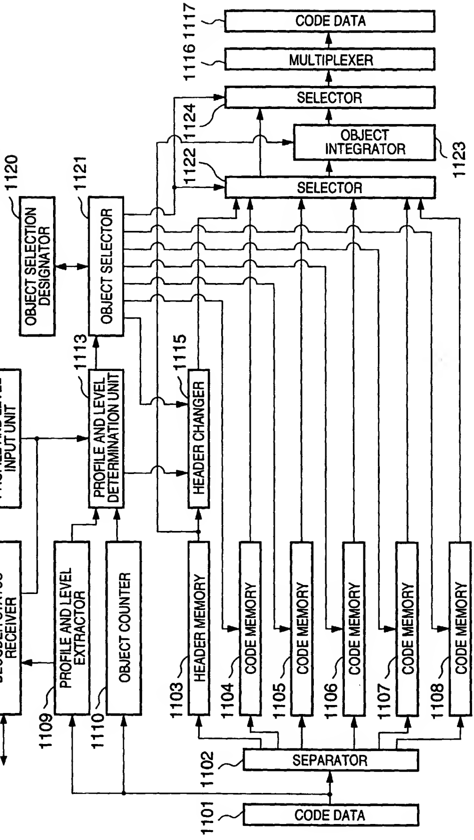

CODE DATA
SEPARATOR
HEADER MEMORY
CODE MEMORY
CODE MEMORY
CODE MEMORY
CODE MEMORY
CODE MEMORY
DECODER STATUS RECEIVER
PROFILE AND LEVEL EXTRACTOR
OBJECT COUNTER
PROFILE AND LEVEL INPUT UNIT
PROFILE AND LEVEL DETERMINATION UNIT
HEADER CHANGER
OBJECT SELECTION DESIGNATOR
OBJECT SELECTOR
SELECTOR
OBJECT INTEGRATOR
SELECTOR
MULTIPLEXER
CODE DATA
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1115
1116
1117
1120
1121
1122
1123
1124

FIG. 34

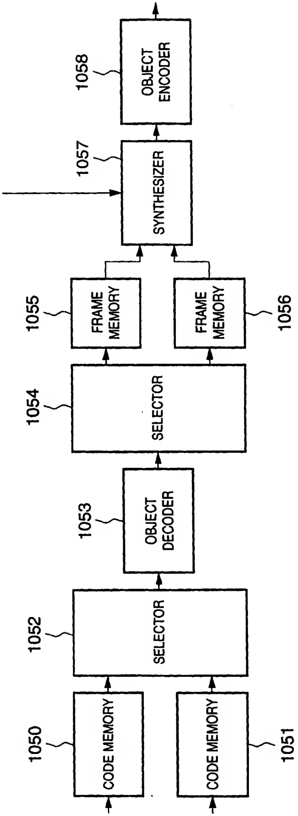
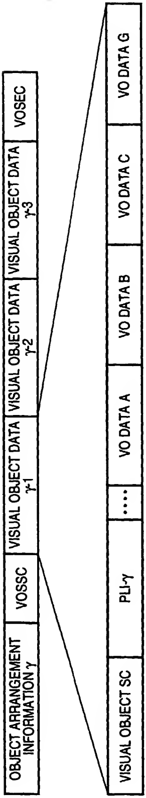

1050
CODE MEMORY
1051
CODE MEMORY
1052
SELECTOR
1053
OBJECT DECODER
1054
SELECTOR
1055
FRAME MEMORY
1056
FRAME MEMORY
1057
SYNTHESIZER
1058
OBJECT ENCODER

FIG. 35


OBJECT ARRANGEMENT INFORMATION γ
VOSSC
VISUAL OBJECT DATA γ-1
VISUAL OBJECT DATA γ-2
VISUAL OBJECT DATA γ-3
VOSEC
VISUAL OBJECT SC
PLI-γ
....
VO DATA A
VO DATA B
VO DATA C
VO DATA G

F I G. 36

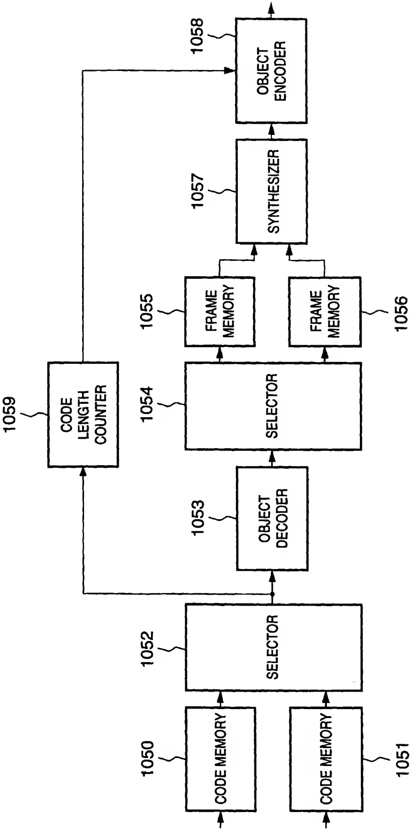

1050
CODE MEMORY
1051
CODE MEMORY
1052
SELECTOR
1053
OBJECT DECODER
1054
SELECTOR
1055
FRAME MEMORY
1056
FRAME MEMORY
1057
SYNTHESIZER
1058
OBJECT ENCODER
1059
CODE LENGTH COUNTER

FIG. 37

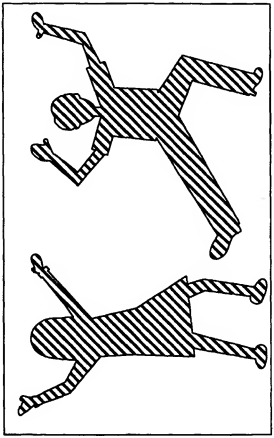


FIG. 38

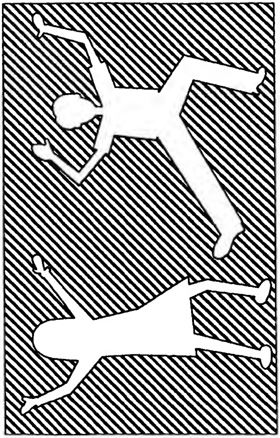


FIG. 39

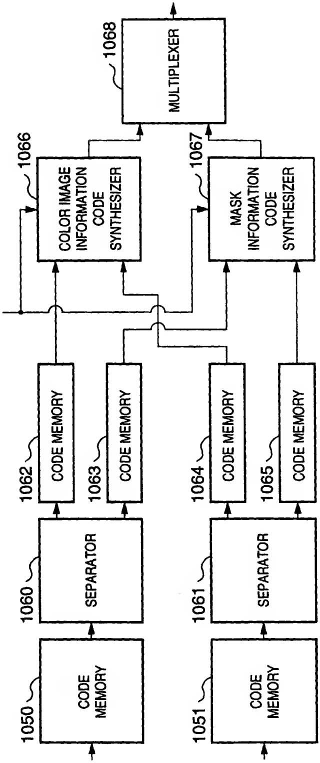

1050
CODE MEMORY
1060
SEPARATOR
1062
CODE MEMORY
1063
CODE MEMORY
1051
CODE MEMORY
1061
SEPARATOR
1064
CODE MEMORY
1065
CODE MEMORY
1066
COLOR IMAGE INFORMATION CODE SYNTHESIZER
1067
MASK INFORMATION CODE SYNTHESIZER
1068
MULTIPLEXER

FIG. 40

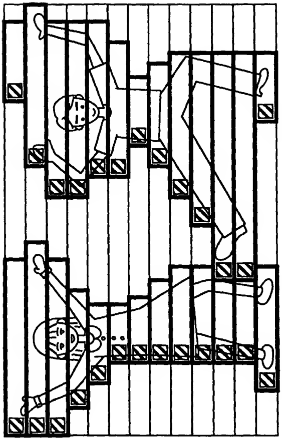


FIG. 41

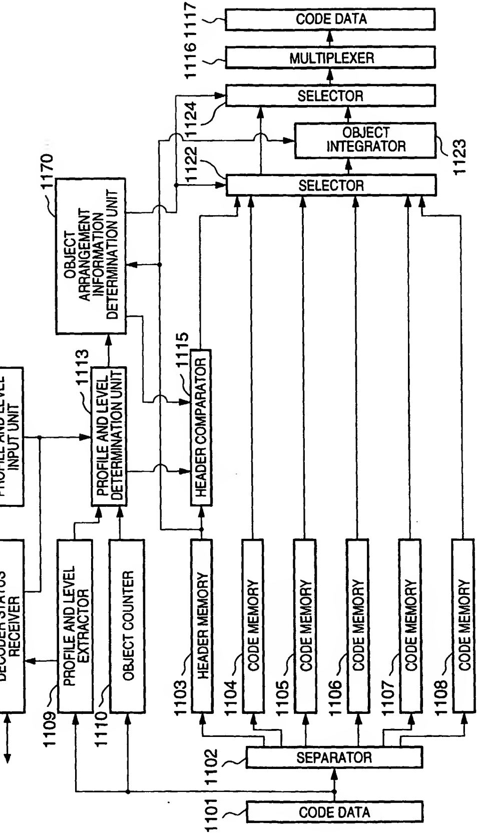

CODE DATA
1101
SEPARATOR
1102
HEADER MEMORY
1103
CODE MEMORY
1104
CODE MEMORY
1105
CODE MEMORY
1106
CODE MEMORY
1107
CODE MEMORY
1108
PROFILE AND LEVEL EXTRACTOR
1109
OBJECT COUNTER
1110
DECODER STATUS RECEIVER
1111
PROFILE AND LEVEL INPUT UNIT
1112
PROFILE AND LEVEL DETERMINATION UNIT
1113
HEADER COMPARATOR
1115
OBJECT ARRANGEMENT INFORMATION DETERMINATION UNIT
1170
SELECTOR
1122
OBJECT INTEGRATOR
1123
SELECTOR
1124
MULTIPLEXER
1116
CODE DATA
1117

FIG. 42

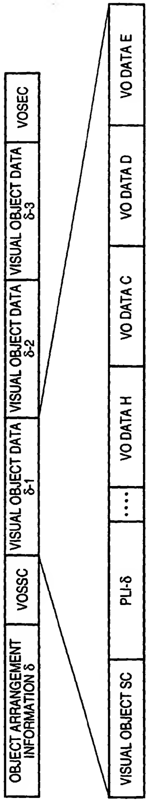

OBJECT ARRANGEMENT INFORMATION δ
VOSSC
VISUAL OBJECT DATA δ-1
VISUAL OBJECT DATA δ-2
VISUAL OBJECT DATA δ-3
VOSEC
VISUAL OBJECT SC
PLI-δ
....
VO DATA H
VO DATA C
VO DATA D
VO DATA E

FIG. 43

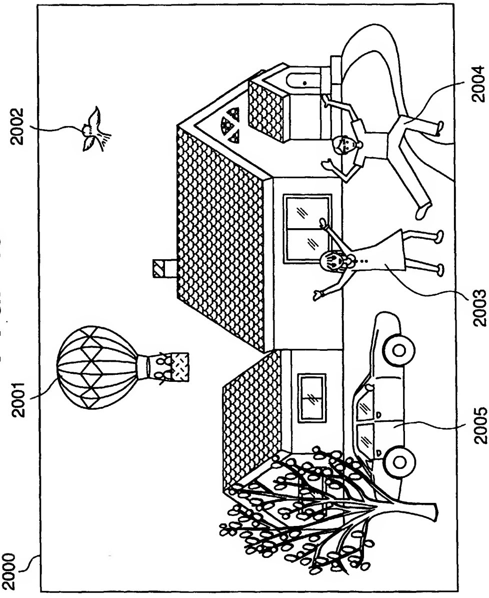


FIG. 43

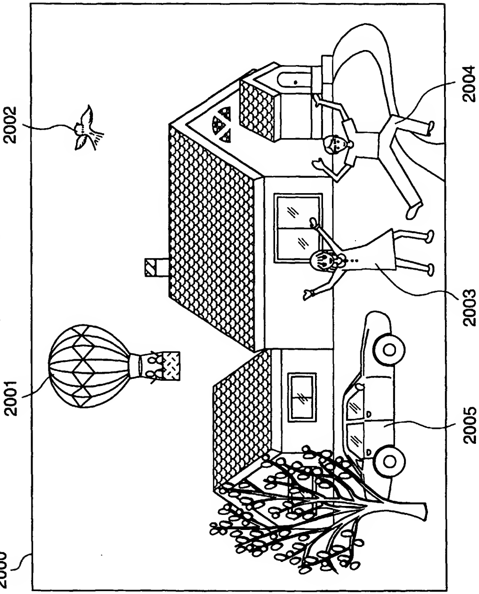


FIG. 44

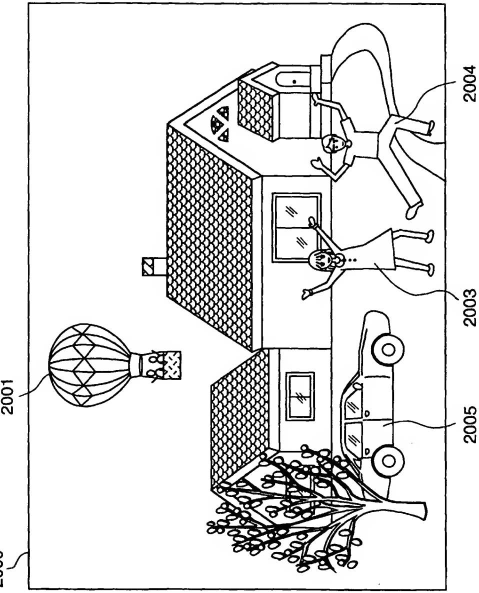


FIG. 45

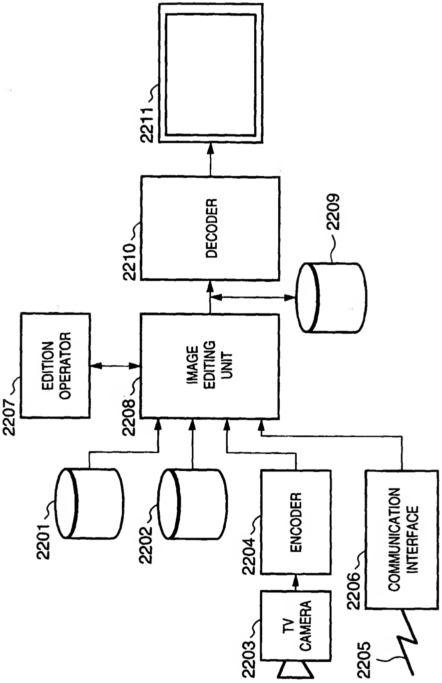

2211
DECODER
2210
2209
EDITION OPERATOR
2207
IMAGE EDITING UNIT
2208
2201
2202
ENCODER
2204
COMMUNICATION INTERFACE
2206
TV CAMERA
2203
2205

FIG. 46A

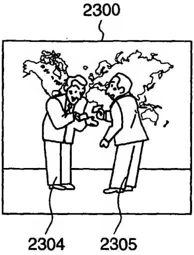


FIG. 46B

FIG. 46C

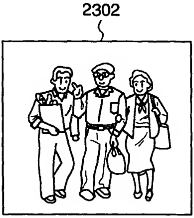


FIG. 46D

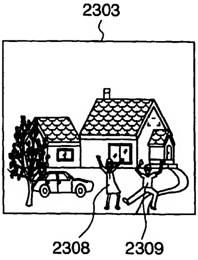


FIG. 47

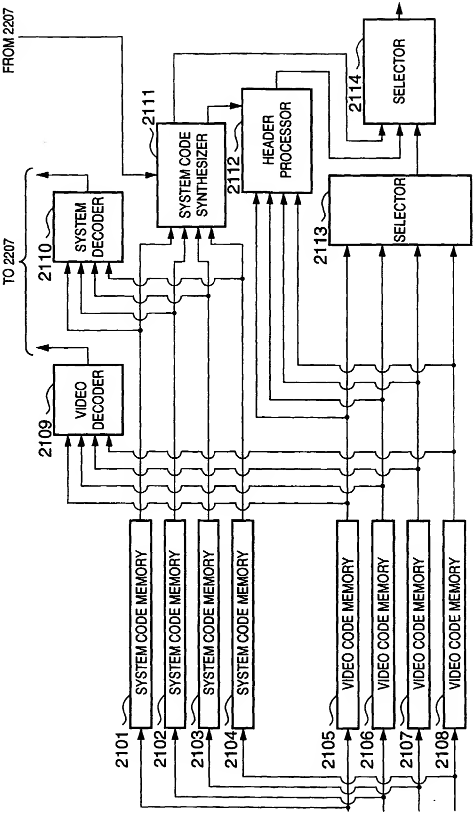

FROM 2207
TO 2207
2101
2102
2103
2104
2105
2106
2107
2108
SYSTEM CODE MEMORY
SYSTEM CODE MEMORY
SYSTEM CODE MEMORY
SYSTEM CODE MEMORY
VIDEO CODE MEMORY
VIDEO CODE MEMORY
VIDEO CODE MEMORY
VIDEO CODE MEMORY
2109
VIDEO DECODER
2110
SYSTEM DECODER
2111
SYSTEM CODE SYNTHESIZER
2112
HEADER PROCESSOR
2113
SELECTOR
2114
SELECTOR

FIG. 48

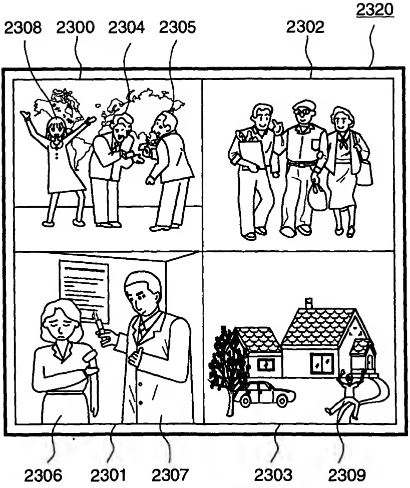


FIG. 49

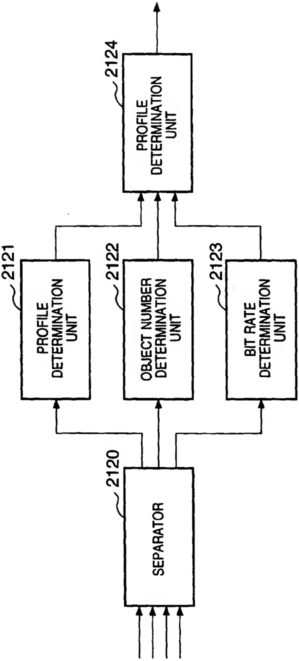

2120
SEPARATOR
2121
PROFILE DETERMINATION UNIT
2122
OBJECT NUMBER DETERMINATION UNIT
2123
BIT RATE DETERMINATION UNIT
2124
PROFILE DETERMINATION UNIT

FIG. 50A

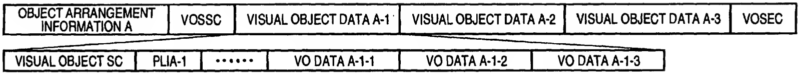


FIG. 50B

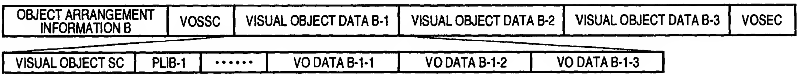


FIG. 50C

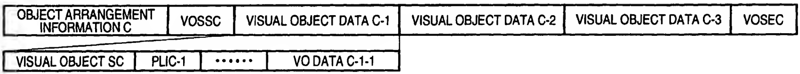


FIG. 50D

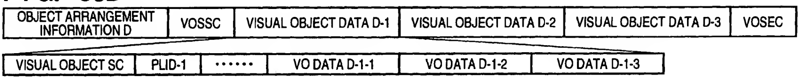


FIG. 50E

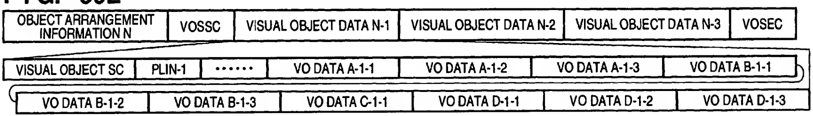


FIG. 51

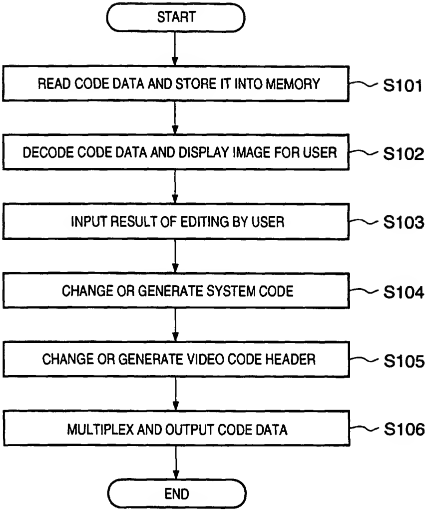

START
READ CODE DATA AND STORE IT INTO MEMORY
S101
DECODE CODE DATA AND DISPLAY IMAGE FOR USER
S102
INPUT RESULT OF EDITING BY USER
S103
CHANGE OR GENERATE SYSTEM CODE
S104
CHANGE OR GENERATE VIDEO CODE HEADER
S105
MULTIPLEX AND OUTPUT CODE DATA
S106
END

FIG. 52

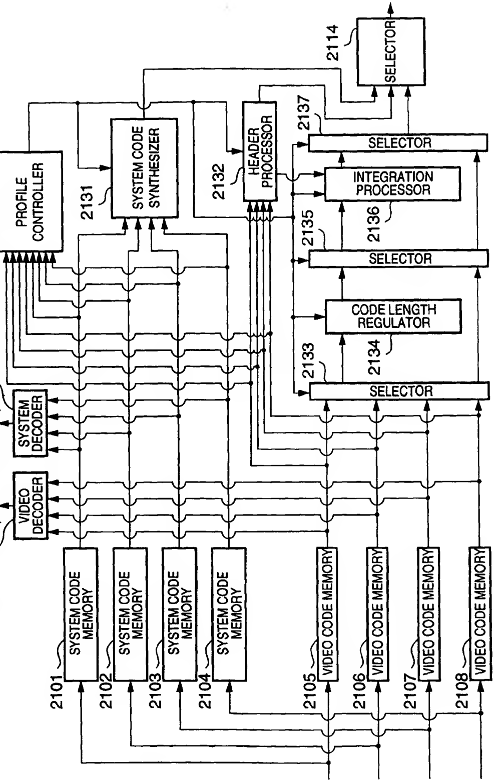

SYSTEM CODE MEMORY
VIDEO CODE MEMORY
2101
2102
2103
2104
2105
2106
2107
2108
2109
VIDEO DECODER
2110
SYSTEM DECODER
TO 2207
2130
PROFILE CONTROLLER
FROM 2207
2131
SYSTEM CODE SYNTHESIZER
2132
HEADER PROCESSOR
2133
SELECTOR
2134
CODE LENGTH REGULATOR
2135
SELECTOR
2136
INTEGRATION PROCESSOR
2137
SELECTOR
2114
SELECTOR

FIG. 53A

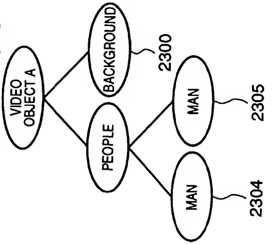


FIG. 53B

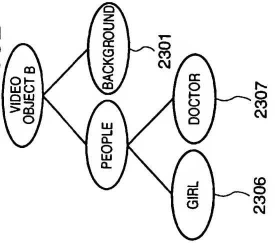


FIG. 53C

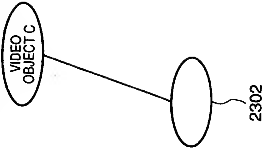


FIG. 53D

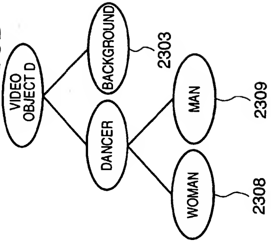


FIG. 54

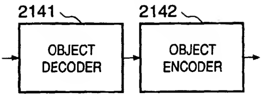


FIG. 55

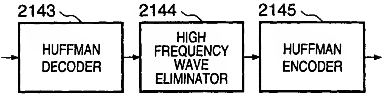


FIG. 56

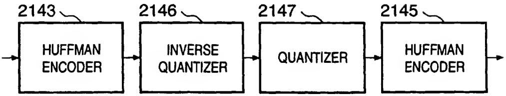
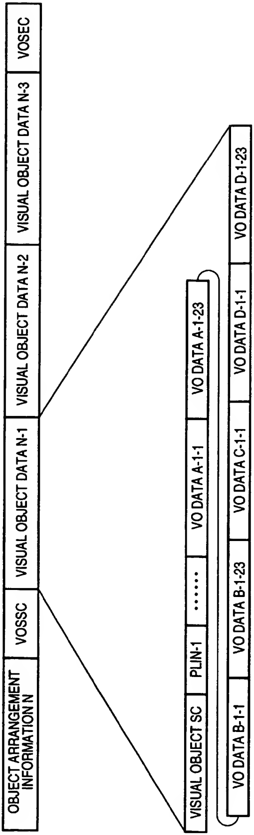


FIG. 57


OBJECT ARRANGEMENT INFORMATION N
VOSSC
VISUAL OBJECT DATA N-1
VISUAL OBJECT DATA N-2
VISUAL OBJECT DATA N-3
VOSEC
VISUAL OBJECT SC
PLIN-1
....
VO DATA A-1-1
VO DATA A-1-23
VO DATA B-1-1
VO DATA B-1-23
VO DATA C-1-1
VO DATA D-1-1
VO DATA D-1-23

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 862 330 A2

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
02.09.1998 Bulletin 1998/36

(51) Int. Cl.[6]: **H04N 7/26**

(21) Application number: **98103443.2**

(22) Date of filing: **27.02.1998**

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **27.02.1997 JP 43217/97**

(71) Applicant:
Matsushita Electric Industrial Co., Ltd.
Kadoma-shi, Osaka 571-8501 (JP)

(72) Inventor: **Takahashi, Toshiya**
567-0832 (JP)

(74) Representative:
Kügele, Bernhard et al
NOVAPAT INTERNATIONAL SA,
9, Rue du Valais
1202 Genève (CH)

(54) **Object data processing apparatus, object data recording apparatus, data storage media, data structure for transmission**

(57) An object data processing apparatus for decoding N pieces of coded data (N = positive integer) obtained by compressively coding N pieces of object data which constitute individual data to be recorded or transmitted and have a hierarchical structure, for each object data. This apparatus includes hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the N pieces of object data, according to the coded data; and table creation means for creating, according to the hierarchical information, an object table on which the respective object data are correlated with coded data of the respective object data. Therefore, the apparatus can perform extraction, selection, or retrieval of coded data corresponding to a specific object at high speed, and this enables the user to edit or replace the object data in short time with high controllability.

EP 0 862 330 A2

Description

FIELD OF THE INVENTION

The present invention relates to object data processing apparatus, object data recording apparatus, data storage media, and data structure for transmission. More particularly, the invention relates to an apparatus for decoding compressed data, such as compressed digital video data, digital audio data, and program data, an apparatus for selecting desired data from the compressed data, an apparatus for recording the compressed data, a medium storing the compressed data, an apparatus for outputting the compressed data, and a data structure for transmitting the compressed data.

BACKGROUND OF THE INVENTION

In recent years, with the progress in information compression technology, a digital video/audio service providing video information and audio information by digital signals has been put to practical use for broadcasting media, such as ground broadcasting, satellite broadcasting, and CATV.

Under the existing circumstances, as a compressive coding method for the next generation, an object coding method has attracted attention. This object coding method is not to uniformly compress the whole image, i.e., video data corresponding to a single image, but to compress video data corresponding to a single image in units of individual objects constituting the image while paying attention to the contents of the image.

When video data corresponding to a single image is subjected to the compressive coding in object units, compressed (coded) video data is separable corresponding to the respective objects, whereby a specific object in the image can be extracted or replaced.

Meanwhile, as a method of implementing a data transmission format for making the best use of the object coding method, a method of multiplexing compressed video data, audio data, and other digital data is discussed.

There is MPEG4 as an international standard of a method of multiplexing data compressed by the object coding method (ISO/IEC JTC1/SC29WG11 N1483, "System Working Draft", November 1996). Hereinafter, a description is given of the data multiplexing method based on MPEG 4 and a method for reproducing the multiplexed data, with reference to figures.

Figure 18 is a diagram for explaining the object coding method. In the figure, reference numeral 120 designates a scene (an image) in a series of images obtained from video data with audio. This scene 120 is composed of a plurality of objects (sub-images) making a hierarchical structure. To be specific, the scene 120 is composed of three objects: a background image (background) 121, a moving object 122 that moves in the background, and a background audio 123 attendant on the background. The moving object 122 is composed of four objects: a first wheel 124, a second wheel 125, a body 126, and a moving object audio 127 attendant on the moving object. Further, the object of body 126 is composed of two objects: a window 128 and the other part 129. In the hierarchical structure, the objects 121~123 belong to the uppermost first layer L1, the objects 124~127 belong to the second layer L2 lower than the first layer L1, and the objects 128 and 129 belong to the third layer L3 lower than the second layer L2.

In the object coding method, scene data corresponding to the scene 120 are compressively coded in units of the lowermost objects constituting the scene 120. In other words, scene data corresponding to the scene 120 are compressively coded for each of the objects 121, 123, 124, 125, 127, 128 and 129.

Figure 19 is a diagram for explaining a data structure for transmitting coded data corresponding to the respective objects mentioned above, which is obtained by performing object coding to the scene data of the scene 120.

In figure 19, MEg shows a multiplexed bit stream having a prescribed format, obtained by multiplexing coded data of the respective objects and auxiliary data. This multiplexed bit stream MEg is transmitted as coded data corresponding to the scene data.

The multiplexed bit stream MEg is partitioned into plural packets in prescribed units, i.e., each packet having prescribed number of bytes, and coded data of the respective objects are allocated to the packets having their own values (SLC=1, 2, ...) as logical channels (LC).

To be specific, in the multiplexed bit stream MEg shown in figure 19, coded video data of object [1] is allocated to packets Pa3 and Pa6 having a logical channel SLC=3, coded video data of object [2] is allocated to packets Pa5 and Pa7 having a logical channel SLC=4, and coded audio data of object [3] is allocated to a packet 4 having a logical channel SLC=5. Information relating to the byte number of packet when multiplexed, the logical channel LC of each packet, and the packet transmission order is allocated as control information to a packet having another logical channel (not shown) for transmission.

The objects [1] and [2] are the background image 121 and the moving object 122 shown in figure 18, respectively, and the object [3] is the background audio 123 shown in figure 18.

In the multiplexed bit stream MEg, allocated to the packet Pa1 of logical channel SLC=1 is information relating to a scene composition method for regenerating the scene composed of the respective objects (composition stream), and allocated to the packet Pa2 of logical channel SLC=2 is information showing how the coded data of the respective objects are multiplexed (stream association table).

Accordingly, when a plurality of coded data

obtained by object coding are multiplexed and transmitted, with the coded data of the respective objects, the composition stream showing the structure of a scene composed of the objects and the stream association table showing the correlation of the transmitted streams (each stream being a series of coded data corresponding to each object) are transmitted simultaneously.

Figure 20 is a diagram for explaining a scene description according to the composition stream, illustrating a description SD corresponding to the single image (scene) 120 shown in figure 18.

In the scene description SD according to the composition stream, the image 120 is shown by Scene 140, and the fact that the image 120 shown by Scene 140 is composed of the background image 121, the moving object 122, and the background audio 123 is shown by Video(1) 141, Node(1) 142, and Audio(1) 143, respectively. Here, Scene 140, Video(1) 141, Node(1) 142, and Audio(1) 143 are descriptors describing the image 120, the background image 121, the moving object 122, and the background audio 123 shown in figure 18, respectively.

Further, in the scene description SD, the fact that the moving object 122 shown by Node(1) 142 is composed of the first wheel 124, the second wheel 125, the body 126, and the moving object audio 127 is shown by Video(2) 144, Video(3) 145, Node(2) 146, and Audio(2) 147, respectively, which are descriptors corresponding to these objects.

Further, the fact that the body 126 shown by Node(2) 146 is composed of the window 128 and the other part 129 is shown by Video(4) 148 and Video(5) 149, respectively, which are descriptors corresponding to these objects.

Each of the descriptors is given a stream index (stream id) for identifying a stream corresponding to coded data of each object in the multiplexed bit stream MEg. To be specific, as shown in figure 20, stream indices Sid=1~Sid=5 are given to the descriptors 141~145, respectively, and stream indices Sid=6, Sid=7, and Sid=8 are given to the descriptors 148, 149, and 147, respectively. Sid is a specific number of each stream id.

Accordingly, it can be seen from the scene description SD according to the composition stream that a scene is composed of what kinds of objects. However, the scene description SD according to the composition stream does not describe how the coded data corresponding to the respective objects are multiplexed in the actual multiplexed bit stream MEg.

Figure 21 is a diagram for explaining the stream association table AT.

The stream association table AT shows the relationship between the stream corresponding to coded data of each object (i.e., a series of coded data corresponding to each object) and the logical channel (LC) specifying each packet which is the partition unit of coded data when multiplexed. To be specific, on this table AT, the stream indices (id) of the respective streams, the logical channel values (LC) corresponding to the respective streams, and the logical channel values (LC) corresponding to upper streams of the respective streams are correlated with each other. Here, the logical channel LC corresponding to the upper stream of the streams (Sid=1~3) corresponding to the objects 121~123 of the first layer L1 corresponds to the logical channel LC (SLC=2) of the packet Pa2 to which the stream association table is allocated.

Accordingly, with reference to this table AT, the logical channel LC corresponding to each stream and the logical channel LC of its upper stream (host stream) can be specified.

As described above, since the stream indices (Sid) are added to the descriptors 141~145 and 147~149 of the respective objects in the scene description SD according to the composition stream shown in figure 20, the respective objects can be identified by the stream indices (Sid) from the composition stream and, therefore, the composition stream can be correlated with the stream association table shown in figure 21.

As described above, the multiplexed bit stream MEg includes the composition stream and the stream association table together with the coded data corresponding to the respective objects. Therefore, when the coded data of the respective objects are reproduced by decoding according to the multiplexed bit stream MEg, it is possible to extract or retrieve coded data of a specific object designated according to the composition stream and the stream association table. This enables, for example, edition of the objects 121 to 129 constituting the scene 120 on the reproduction end.

In the multiplexed bit stream format according to the prior art object coding, the scene description is expressed as information (composition stream) separated from information relating to the multiplexed state of the respective coded data and the logical channels corresponding to the respective streams (stream association table). The reason is as follows. In order to realize exchange of the contents of streams corresponding to the respective objects and to facilitate interface between the multiplexed bit stream and applications treating this multiplexed bit stream without changing the scene composition (i.e., the hierarchical structure of the objects constituting a scene), the structure for multiplexing, which depends on the physical layer of the multiplexed bit stream, must be separated from main information (coded data) included in the multiplexed bit stream.

However, the multiplexed bit stream format according to the prior art has the following drawbacks.

A great advantage of object coding resides in that it enables extraction of coded data of a specific object from the multiplexed bit stream, and retrieval of a specific object on the data base containing the multiplexed bit stream.

However, in order to recognize coded data of individual objects from the multiplexed bit stream MEg of

the above-mentioned structure, a complicated procedure is required as follows. For example, to recognize coded data of lower-layer objects from plural objects having a hierarchical structure, initially, the scene description according to the composition stream included in the multiplexed bit stream MEg is interpreted to find an object corresponding to a node, and a stream corresponding to a lower object being a component of the object (node) is specified. Then, the stream association table AT is interpreted and, according to the stream id of the specified stream, a logical channel LC corresponding to the stream id is found. Thereby, coded data of the specified object can be extracted from the multiplexed bit stream MEg.

Furthermore, since the hierarchical relationship of the streams corresponding to the respective objects can be seen from the stream association table AT, it is possible to analogize coded data of a specific object according to the stream association table AT alone, but this analogy takes time and is not reliable.

That is, on the stream association table AT, information relating to objects as nodes is not clearly defined. In addition, since this table AT does not show the type of stream corresponding to coded data (for example, whether a stream corresponds to video data or audio data), other information such as the composition stream should be referred to. Further, for each stream, only its upper stream is known from the table AT. So, it is impossible to uniquely know that coded data of each object is composed of which stream, and interpretation takes time.

For example, in the scene description SD according to the composition stream shown in figure 20, although Node(2) 146 corresponding to the object 126 exists, a stream corresponding to Node(2) does not exist. So, on the stream association table AT shown in figure 21, an entry corresponding to Node(2) (i.e., stream id, LC corresponding to the stream, and LC corresponding to the stream's upper-layer stream) does not exists.

Accordingly, in order to extract the object 126 corresponding to Node(2), initially, stream indices (id) corresponding to the lower-layer objects 128 and 129 of Node(2) 146 must be decided on the basis of the scene description SD according to the composition stream (refer to figure 20) and, thereafter, the logical channels (LC) of packets containing the streams having the decided stream indices must be defined on the stream association table AT (refer to figure 21).

Further, there is a case where coded data corresponding to plural objects are transmitted without being multiplexed in a particular transmission medium, such as computer network (internet). In this case, the bit stream has a data structure including no logical channels, and does not include the stream association table.

In this case, detection of a specific object from the bit stream is carried out by interpreting the hierarchical structure of the objects on the basis of the scene description SD according to the composition stream. However, when the number of the objects increases considerably, it requires a lot of time to interpret the hierarchical structure of the objects on the basis of the composition stream, resulting in poor controllability in replacement or edition of objects in a scene.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide an object data processing apparatus that speedily performs extraction, selection, or retrieval of coded data corresponding to a specific object from coded data corresponding to plural objects, whereby the user can edit or replace object data in short time with high controllability.

It is another object of the present invention to provide an object data recording apparatus that records coded data corresponding to plural objects so that coded data of a specific object among the plural objects can be extracted, selected, or retrieved easily and speedily.

It is still another object of the present invention to provide a data storage medium containing coded data having a data structure, which data structure realizes simple and speedy extraction, selection, or retrieval of coded data of a specific object from coded data corresponding to plural objects.

It is a further object of the present invention to provide a data structure for transmission that realizes simple and speedy extraction, selection, or retrieval of coded data of a specific object from coded data corresponding to plural objects.

Other objects and advantages of the invention will become apparent from the detailed description that follows. The detailed description and specific embodiments described are provided only for illustration since various additions and modifications within the scope of the invention will be apparent to those of skill in the art from the detailed description.

According to a first aspect of the present invention, there is provided an object data processing apparatus for decoding N pieces of coded data (N = positive integer) obtained by compressively coding N pieces of object data which constitute individual data to be recorded or transmitted and have a hierarchical structure, for each object data. This apparatus includes hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the N pieces of object data, according to the coded data; and table creation means for creating, according to the hierarchical information, an object table on which the respective object data are correlated with coded data corresponding to the respective object data. Therefore, extraction, selection or retrieval of coded data of a specific object can be carried out easily and speedily, and this enables the user to edit or replace the object data in short time with high controllability.

According to a second aspect of the present inven-

tion, there is provided an object data processing apparatus for decoding N pieces of coded data (N = positive integer) obtained by compressively coding scene data corresponding to one scene, for each of N pieces of objects constituting the scene. This apparatus includes hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the respective objects constituting the scene, according to the coded data; and table creation means for creating, according to the hierarchical information, an object table on which the respective objects are correlated with coded data corresponding to the respective objects. Therefore, on the decoder side, extraction, selection or retrieval of a specific object from plural objects (video and audio) constituting one scene can be carried out easily and speedily with reference to the object table.

According to a third aspect of the present invention, in the object data processing apparatus according to the second aspect, the hierarchical information extraction means is constructed so that it extracts priority information showing the priority order of the respective objects, according to the coded data, in addition to the hierarchical information; and the table creation means is constructed so that it creates, according to the hierarchical information and the priority information, an object table on which the respective objects are correlated with coded data corresponding to the respective objects and the priority order of the respective objects are shown. Therefore, when the throughput of the decoding apparatus is low and the apparatus cannot decode coded data of all objects, only objects having priorities higher than a prescribed priority are subjected to decoding.

According to a fourth aspect of the present invention, the object data processing apparatus according to the second aspect further includes identification information detection means for detecting identification information for identifying coded data of a specific object designated, with reference to the object table; and decoding means for extracting coded data of the specific object from the N pieces of coded data according to the identification information, and decoding the extracted coded data. Therefore, in the decoding apparatus, retrieval of an object specified by the user is facilitated.

According to a fifth aspect of the present invention, there is provided an object data processing apparatus for decoding multiplexed data including N pieces of coded data (N = positive integer) obtained by compressively coding scene data corresponding to one scene, for each of N pieces of objects constituting the scene. This apparatus includes hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the N pieces of objects constituting the scene, according to information showing the correlation of the respective coded data and included in the multiplexed data; and table creation means for creating, according to the hierarchical information, an object table on which the respective objects are correlated with coded data corresponding to the respective objects. Therefore, extraction, selection or retrieval of a specific object from plural objects (video and audio) constituting one scene can be carried out easily and speedily, on the basis of the multiplexed data, with reference to the object table.

According to a sixth aspect of the present invention, in the object data processing apparatus according to the fifth aspect, the hierarchical information extraction means is constructed so that it extracts priority information showing the priority order of the respective objects, according to the multiplexed data, in addition to the hierarchical information; and the table creation means is constructed so that it creates, according to the hierarchical information and the priority information, an object table on which the respective objects are correlated with coded data corresponding to the respective objects and the priority order of the respective objects are shown. Therefore, when the throughput of the decoding apparatus is low and the apparatus cannot decode coded data of all objects, only objects having priorities higher than a prescribed priority are subjected to decoding.

According to a seventh aspect of the present invention, the object data processing apparatus according to the fifth aspect further includes identification information detection means for detecting identification information for identifying coded data of a specific object designated, with reference to the object table; and decoding means for extracting coded data of the specific object from the multiplexed data according to the identification information, and decoding the extracted coded data. Therefore, in the decoding apparatus, retrieval of an object specified by the user is facilitated.

According to an eighth aspect of the present invention, there is provided an object data processing apparatus for selecting coded data of a specific object data from N pieces of coded data (N = positive integer) obtained by compressively coding N pieces of object data which constitute individual data to be recorded or transmitted and have a hierarchical structure, for each object data. This apparatus includes hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the N pieces of object data, according to the coded data; and table creation means for creating, according to the hierarchical information, an object table on which the respective object data are correlated with coded data corresponding to the respective object data. This apparatus selects coded data of a specific object data from the N pieces of coded data with reference to the object table and outputting the selected coded data. Therefore, selection of coded data of a specific object can be carried out easily and speedily, and this enables the user to edit, replace, or retrieve the object data in short time with high controllability.

According to a ninth aspect of the present invention, there is provided an object data processing appa-

ratus for selecting coded data of a specific object from N pieces of coded data (N = positive integer) obtained by compressively coding scene data corresponding to one scene, for each of N pieces of objects constituting the scene. This apparatus includes hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the respective objects constituting the scene, according to the coded data; and table creation means for creating, according to the hierarchical information, an object table on which the respective objects are correlated with coded data corresponding to the respective objects. This apparatus selects coded data of a specific object from the N pieces of coded data with reference to the object table and outputting the selected coded data. Therefore, on the decoder side, selection of a specific object from plural objects (video and audio) constituting one scene can be carried out easily and speedily.

According to a tenth aspect of the present invention, there is provided an object data processing apparatus for selecting coded data of a specific object from multiplexed data including N pieces of coded data (N = positive integer) obtained by compressively coding scene data corresponding to one scene, for each of N pieces of objects constituting the scene. This apparatus includes hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the respective objects constituting the scene, according to information showing the correlation of the respective coded data and included in the multiplexed data; and table creation means for creating, according to the hierarchical information, an object table on which the respective objects are correlated with coded data corresponding to the respective objects. This apparatus selects coded data of a specific object from the multiplexed data with reference to the object table and outputting the selected coded data. Therefore, on the basis of the multiplexed data, selection of a specific object from plural objects (video and audio) constituting one scene can be carried out easily and speedily.

According to an eleventh object of the present invention, there is provided an object data recording apparatus having a data storage for storing data, and recording N pieces of coded data (N = positive integer) in the data storage, which coded data are obtained by compressively coding N pieces of object data which constitute individual data to be recorded or transmitted and have a hierarchical structure, for each object data. This apparatus includes hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the respective object data, according to the coded data; and table creation means for creating, according to the hierarchical information, an object table on which the respective object data are correlated with coded data corresponding to the respective object data. This apparatus records the N pieces of coded data and the object table corresponding to these coded data in the data storage. Therefore, extraction, selection or retrieval of coded data of a specific object can be carried out easily and speedily with reference to the object table.

According to a twelfth aspect of the present invention, there is provided an object data recording apparatus having a data storage for storing data, and recording N pieces of coded data (N = positive integer) in the data storage, which coded data are obtained by compressively coding scene data corresponding to one scene, for each of N pieces of objects constituting the scene. This apparatus includes hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the respective objects constituting the scene, according to the coded data; and table creation means for creating, according to the hierarchical information, an object table on which the respective objects are correlated with coded data corresponding to the respective objects. This apparatus records the N pieces of coded data and the object table corresponding to these coded data in the data storage. Therefore, on the decoder side, extraction, selection or retrieval of a specific object from plural objects (video and audio) constituting one scene can be performed easily and speedily with reference to the object table.

According to a thirteenth aspect of the present invention, in the object data recording apparatus according to the twelfth aspect, the object table corresponding to the N pieces of coded data is added to the N pieces of coded data when being recorded. Therefore, management of the recorded object table is facilitated.

According to a fourteenth aspect of the present invention, in the object data recording apparatus according to the twelfth aspect, the object table corresponding to the N pieces of coded data is separated from the N pieces of coded data when being recorded. Therefore, regardless of the size of the coded data, recording of the object table to a storage medium is carried out with high reliability.

According to a fifteenth aspect of the present invention, there is provided an object data recording apparatus having a data storage for storing data, and recording multiplexed data including N pieces of coded data (N = positive integer) in the data storage, which coded data are obtained by compressively coding scene data corresponding to one scene, for each of N pieces of objects constituting the scene. This apparatus includes hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the respective objects constituting the scene, according to information showing the correlation of the respective coded data and included in the multiplexed data; and table creation means for creating, according to the hierarchical information, an object table on which the respective objects are correlated with coded data corresponding to the respective objects. This apparatus records the multiplexed data and the object table corresponding to the multiplexed data in the

data storage. Therefore, on the decoder side, extraction, selection or retrieval of a specific object from plural objects (video and audio) constituting one scene can be performed easily and speedily with reference to the object table.

According to a sixteenth aspect of the present invention, in the object data recording apparatus according to the fifteenth aspect, the object table corresponding to the multiplexed data is added to the multiplexed data when being recorded. Therefore, management of the recorded object table is facilitated.

According to a seventeenth aspect of the present invention, in the object data recording apparatus according to the fifteenth aspect, the object table corresponding to the multiplexed data is separated from the multiplexed data when being recorded. Therefore, regardless of the size of the multiplexed data, recording of the object table in a storage medium is carried out with high reliability.

According to an eighteenth aspect of the present invention, there is provided a data storage medium containing relevant data relating to individual data to be recorded or transmitted, wherein the relevant data includes an object table on which N pieces of object data (N = positive integer) constituting the individual data and having a hierarchical structure are correlated with N pieces of coded data obtained by compressively coding the respective object data. Therefore, extraction, selection or retrieval of specific object data from the individual data can be carried out easily and speedily with reference to the object table.

According to a nineteenth aspect of the present invention, there is provided a data storage medium containing relevant data corresponding to one scene, wherein the relevant data includes an object table on which N pieces of coded data (N = positive integer) obtained by compressively coding scene data corresponding to one scene for each of N pieces of objects constituting the scene are correlated with the respective objects. Therefore, on the decoder side, extraction, selection or retrieval of a specific object from plural objects (video and audio) constituting one scene can be performed easily and speedily with reference to the object table.

According to a twentieth aspect of the present invention, there is provided an object data processing apparatus for outputting N pieces of coded data (N = positive integer) obtained by compressively coding N pieces of object data which constitute individual data to be recorded or transmitted and have a hierarchical structure, for each object data. This apparatus includes hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the N pieces of object data, according to the coded data; and table creation means for creating, according to the hierarchical information, an object table on which the respective object data are correlated with coded data corresponding to the respective object data. This apparatus outputs the N pieces of coded data to which the object table corresponding to these coded data is added. Therefore, it is not necessary to create an object table on the decoder side, whereby edition, replacement or retrieval of object data can be performed by a simple structure, in short time with high controllability.

According to a twenty-first aspect of the present invention, there is provided an object data processing apparatus for outputting N pieces of coded data (N = positive integer) obtained by compressively coding scene data corresponding to one scene, for each of N pieces of objects constituting the scene. This apparatus includes hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the respective objects constituting the scene, according to the coded data; and table creation means for creating, according to the hierarchical information, an object table on which the respective objects are correlated with coded data corresponding to the respective objects. This apparatus outputs the N pieces of coded data to which the object table corresponding to these coded data is added. Therefore, it is not necessary to create an object table on the decoder side, whereby edition, replacement or retrieval of objects (video and audio) constituting one scene can be performed by a simple structure, in short time with high controllability.

According to a twenty-second aspect of the present invention, there is provided an object data processing apparatus for decoding data output from the object data processing apparatus according to the twenty-first aspect. This apparatus includes data separation means for separating the object table from the output data; and table storage means for storing the separated object table. In this apparatus, decoding of the coded data corresponding to the respective objects is controlled using the information shown in the object table stored in the table storage means. Therefore, it is possible to realize a decoding apparatus of simple structure that can perform edition, replacement or retrieval of objects (video and audio) constituting one scene in short time with high controllability.

According to a twenty-third aspect of the present invention, there is provided an object data processing apparatus for outputting multiplexed data including N pieces of coded data (N = positive integer) obtained by compressively coding scene data corresponding to one scene, for each of N pieces of objects constituting the scene. This apparatus includes hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the respective objects constituting the scene, according to information showing the correlation of the respective coded data included in the multiplexed data; and table creation means for creating, according to the hierarchical information, an object table on which the respective objects are correlated with coded data corresponding to the

respective objects. This apparatus outputs the multiplexed data to which the object table corresponding to the multiplexed data is added. Therefore, it is not necessary to create an object table on the decoder side, whereby edition, replacement or retrieval of objects (video and audio) constituting one scene can be performed by a simple structure, in short time with high controllability.

According to a twenty-fourth aspect of the present invention, there is provided an object data processing apparatus for decoding data output from the object data processing apparatus according to the twenty-third aspect. This apparatus includes data separation means for separating the object table from the output data; and table storage means for storing the separated object table. In this apparatus, decoding of the coded data corresponding to the respective objects is controlled using the information shown in the object table stored in the table storage means. Therefore, it is possible to realize a decoding apparatus of simple structure that can perform edition, replacement or retrieval of objects (video and audio) constituting one scene in short time with high controllability.

According to a twenty-fifth aspect of the present invention, there is provided a data structure for transmitting N pieces of coded data (N = positive integer) obtained by compressively coding N pieces of object data which constitute individual data to be recorded or transmitted and have a hierarchical structure, for each object data. In this data structure, a data group comprising the N pieces of coded data includes an object table on which the respective object data are correlated with coded data corresponding to the respective object data. Therefore, extraction, selection or retrieval of coded data corresponding to a specific object can be carried out easily and speedily with reference to the object table.

According to a twenty-sixth aspect of the present invention, there is provided a data structure for transmitting N pieces of coded data (N = positive integer) obtained by compressively coding scene data corresponding to one scene, for each of N pieces of objects constituting the scene. In this data structure, a data group comprising the N pieces of coded data includes an object table on which the respective objects are correlated with coded data corresponding to the respective objects. Therefore, on the decoder side, extraction, selection or retrieval of a specific objected from plural objects (video and audio) constituting one scene can be carried out easily and speedily with reference to the object table.

According to a twenty-seventh aspect of the present invention, there is provided an object data processing apparatus for processing multiplexed data including N pieces of coded data (N = positive integer) and being partitioned into plural packets each having a prescribed code quantity, which coded data are obtained by compressively coding N pieces of object data which constitute individual data to be recorded or transmitted and have a hierarchical structure, for each object data. This apparatus includes hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the N pieces of object data, according to information showing the correlation of the respective coded data and included in the multiplexed data; and table creation means for creating, according to the hierarchical information, an object table showing the hierarchical relationship of the plural packets constituting the multiplexed data. Therefore, extraction, selection or retrieval of coded data corresponding to a specific object on the basis of the multiplexed data can be carried out easily and speedily with reference to the object table, and this enables the user to edit or replace the object data in short time with high controllability.

According to a twenty-eighth aspect of the present invention, there is provided an object data processing apparatus for processing multiplexed data including N pieces of coded data (N = positive integer) and being partitioned into plural packets each having a prescribed code quantity, which coded data are obtained by compressively coding scene data corresponding to one scene, for each of N pieces of objects constituting the scene. This apparatus includes hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the respective objects constituting the scene, according to information showing correlation of the respective coded data included in the multiplexed data; and table creation means for creating, according to the hierarchical information, an object table showing the hierarchical relationship of the plural packets constituting the multiplexed data. Therefore, extraction, selection or retrieval of a specific object from plural objects (video and audio) constituting one scene on the basis of the multiplexed data can be carried out easily and speedily with reference to the object table.

According to a twenty-ninth aspect of the present invention, there is provided an object data recording apparatus having a data storage for storing data, and recording, in the storage, multiplexed data which includes N pieces of coded data (N = positive integer) and is partitioned into plural packets each packet having a prescribed coded quantity, which coded data are obtained by compressively coding N pieces of object data which constitute individual data to be recorded or transmitted and have a hierarchical structure, for each object data. This apparatus includes hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the N pieces of object data, according to information showing the correlation of the respective coded data and included in the multiplexed data; and table creation means for creating, according to the hierarchical information, an object table showing the hierarchical relationship of the plural packets constituting the

multiplexed data. This apparatus records the multiplexed data and the object table corresponding to the multiplexed data in the data storage. Therefore, extraction, selection or retrieval of coded data corresponding to a specific object can be carried out easily and speedily with reference to the object table.

According to a thirtieth aspect of the present invention, there is provided an object data recording apparatus having a data storage for storing data, and recording, in the data storage, multiplexed data which includes N pieces of coded data (N = positive integer) and is partitioned into plural packets each having a prescribed coded quantity, which coded data are obtained by compressively coding scene data constituting one scene, for each of N pieces of objects constituting the scene. This apparatus includes hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the respective objects constituting the scene, according to information showing the correlation of the respective coded data and included in the multiplexed data; and table creation means for creating, according to the hierarchical information, an object table showing the hierarchical relationship of the plural packets constituting the multiplexed data. This apparatus records the multiplexed data and the object table corresponding to the multiplexed data in the data storage. Therefore, on the decoder side, extraction, selection or retrieval of a specific object from plural objects (video and audio) constituting one scene can be carried out easily and speedily with reference to the object table.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram for explaining an object data decoding apparatus as an object data processing apparatus according to a first embodiment of the present invention.

Figure 2 is a diagram showing an object table created by the object data decoding apparatus according to the first embodiment.

Figure 3 is a flowchart for explaining process steps by a CPU in the object data decoding apparatus according to the first embodiment.

Figure 4 is a diagram for explaining another object table created in the object data decoding apparatus according to the first embodiment, illustrating a table corresponding to upper-layer objects and a table corresponding to lower-layer objects.

Figure 5 is a block diagram for explaining an object data selecting apparatus as an object data processing apparatus according to a second embodiment of the present invention.

Figure 6 is a flowchart for explaining process steps by a CPU in the object data selecting apparatus according to the second embodiment.

Figure 7 is a block diagram for explaining an object data recording apparatus according to a third embodiment of the present invention.

Figure 8 is a flowchart for explaining process steps by a CPU in the object data recording apparatus according to the third embodiment.

Figure 9 is a block diagram for explaining an object data outputting apparatus as an object data processing apparatus according to a fourth embodiment of the present invention.

Figure 10 is a flowchart for explaining process steps by a CPU in the object data outputting apparatus according to the fourth embodiment.

Figure 11 is a diagram for explaining an object data decoding apparatus based on MPEG4 as an object data processing apparatus according to a fifth embodiment of the invention, illustrating the outline of a data transmission system based on MPEG4.

Figure 12 is a block diagram for explaining an object data decoding apparatus as an object data processing apparatus according to the fifth embodiment of the invention.

Figure 13 (a) and (b) is a schematic diagram for explaining an object coding method corresponding to the data transmission system shown in figure 11.

Figure 14(a) is a diagram showing a scene description and figure 14(b) is a diagram showing object descriptors, respectively, used in the data transmission system.

Figure 15 (a) and (b) is a diagram showing an object table obtained from the scene description shown in figure 14(a) and the object descriptors shown in figure 14(b).

Figure 16 is a diagram showing a flow of process steps by a CPU in the object data decoding apparatus according to the fifth embodiment.

Figures 17(a)-17(c) are diagrams for explaining a data storage medium according to the present invention, wherein figure 17(a) shows the structure of a floppy disk, figure 17(b) shows the structure of a floppy disk body, and figure 17(c) shows a computer system using the floppy disk as a storage medium.

Figure 18 is a schematic diagram for explaining an object coding method according to the prior art.

Figure 19 is a diagram showing a data structure of a bit stream obtained by multiplexing data coded by the prior art object coding method and auxiliary data.

Figure 20 is a diagram showing a scene description according to a composition stream included in the bit stream shown in figure 17 as auxiliary data.

Figure 21 is a diagram showing a stream association table included in the bit stream shown in figure 17 as auxiliary data.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[Embodiment 1]

Figure 1 is a block diagram illustrating an object data decoding apparatus as an object data processing apparatus according to a first embodiment of the present invention.

In figure 1, an object data decoding apparatus 101 receives coded data corresponding to a single image (scene), performs decoding of the coded data, and outputs regeneration data obtained by the decoding to the display unit 14. The coded data is identical to the multiplexed bit stream MEg shown in figure 19 in which coded data obtained by object coding of scene data corresponding to a single scene comprising plural objects are multiplexed with auxiliary data. The single scene corresponds to an image of each frame constituting a motion picture. The object data decoding apparatus 101 successively decodes coded data of each frame input thereto, and successively outputs regeneration data corresponding to each frame.

More specifically, the object data decoding apparatus 101 includes demultiplexer 11 and an AV (audio/video) decoder 12. The demultiplexer 11 selects and extracts a composition stream and a stream association table which are auxiliary data Dsub included in the multiplexed bit stream MEg, and outputs coded data Eg corresponding to the respective objects in the multiplexed bit stream MEg, in units of the respective objects, according to a first control signal Cont1. The AV decoder 12 decodes the coded data Eg corresponding to each object according to a second control signal Cont2, and outputs regeneration data Rg corresponding to each scene. Further, the decoding apparatus 101 includes a CPU (central processing unit) 13. The CPU 13 decides a logical channel of a packet containing the coded data Eg of each object according to the stream association table, and supplies the first control signal Cont1 to the demultiplexer 11 according to the result of the decision. Further, the CPU 13 supplies information relating to location of each object in one scene and information relating to display start time of each object, as the second control signal Cont2, to the AV decoder 12, on the basis of a scene description according to the composition stream.

In this first embodiment, the CPU 13 creates an object table showing the correspondences between the respective objects constituting one scene and coded data Eg of the respective objects in the multiplexed bit stream MEg, on the basis of the composition stream and the stream association table, and stores this table in a data storage inside the CPU 13.

Figure 2 is a diagram for explaining an object table T1 corresponding to the scene 120 shown in figure 18.

On this object table T1, various kinds of information are entered, correlated with each object index (object id) which is an index for identifying each object being a constituent of the scene. Each object has its own value Oid as its object index.

To be specific, object indices are uniquely given to the respective object descriptors in the scene description shown in figure 20, and entered to the object table T1 in the order of the object id values Oid.

In figure 2, object indices Oid=1 to Oid=5 are given to the descriptors 141 to 145 of the objects 121 to 125 shown in figure 20, respectively, and object indices Oid=61 and Oid=62 are given to the descriptors 148 and 149 of the objects 128 and 129 shown in figure 20, respectively. Further, an object index Oid=8 is given to the descriptor 147 of the object 127 shown in figure 20.

On the object table T1, correlated with each object index (id), the following components are entered: a logical channel (LC) corresponding to the object; the stream type (i.e., whether the stream is video or audio); a stream index corresponding to the object; indices of upper and lower objects corresponding to the object; logical channels of the upper and lower objects; the index of an object which shares its logical channel with the object (common object id in figure 2), and the priority order of the respective objects. In figure 2, OLC is a specific value of logical channel LC.

An upper object of each object is an object which belongs to an upper layer in the hierarchical structure than a layer to which each object belongs. A lower object of each object is an object which belongs to a lower layer than a layer to which each object belongs.

To be specific, the uppermost object, i.e., an upper object of the objects belonging to the first layer L1 in figure 18, is the object itself. So, the uppermost object has its object id value Oid=1 and its logical channel value OLC=3 for its upper object id value Oid and its upper object logical channel value OLC. Further, the upper objects of the respective objects belonging to the second layer L2 and the third layer L3 in figure 18 are the objects belonging to the first layer L1 and composed of the objects of the second and third layers L2 and L3. Further, objects having no lower objects (lowermost objects) have lower object id values Oid=0. In this first embodiment, a specific object's being the uppermost object is shown by giving its object id as its upper object id, and a specific object's being the lowermost object is shown by giving Oid=0 as its lower object id. However, a special digit or symbol may be used for describing that a specific object is the uppermost or lowermost object.

From the object table T1 so constructed, it can be seen that the object having Oid=2 as its object id is composed of four objects because "Oid=4, 5, 6, 8" is described as its lower object id.

Further, in figure 2, for the object 128 (129) which is a component of the object 126 corresponding to a node having no stream (in the description of figure 20, Node(2) 146), its object id is given as follows. That is, a digit showing that this is a node common to some objects is given as the upper column of its Oid, and a

digit showing its actual object id value is given as the lower column of the Oid.

Accordingly, from the object table T1, it is found that the object 126 having its object id value Oid=6 is merely a node and there exists no coded data corresponding to this node, and the stream of this object 126 is composed of the streams of the objects 128 and 129 having object ids Oid=61 and Oid=62, i.e., the streams of Sid=6 and Sid=7, respectively.

Further, the object 122 having Oid=2 as its object id corresponds to a node, like the object 126 having Oid=6 as its object id. However, since this object 122 is definitely described on the object table T1, there exists a stream corresponding to this object 122, i.e., the stream of Sid=2.

Information included in a stream corresponding to a node is information common to all objects belonging to the node, for example, composition information peculiar to the objects, such as common system clock, display start timing, decoding start time, and display position, and copyright information.

A description is now given of the operation of the object data decoding apparatus 101.

When the multiplexed bit stream MEg shown in figure 19 is input to the object data decoding apparatus 101, the composition stream and the stream association table (auxiliary data Dsub) are extracted from the multiplexed bit stream MEg and supplied to the CPU 13 by the demultiplexer 11.

In the CPU 13, the correspondences between the respective objects constituting the scene 120 shown in figure 19 and their logical channels LC are recognized and, according to the recognition, a first control signal Cont1 is output to the demultiplexer 11.

In the demultiplexer 11, according to the first control signal Cont1, streams corresponding to the respective objects, which are allocated to plural packets in the multiplexed bit stream, are output to the AV decoder 12 in object units.

At this time, in the CPU 13, information relating to each object's display position and display start time is extracted from the scene description according to the composition stream, and the extracted information is output to the AV decoder 12 as a second control signal Cont2.

In the AV decoder 12, a stream (a series of coded data Eg) corresponding to each object output from the demultiplexer 11 is subjected to decoding. Decoded data corresponding to the respective objects are composited according to the second control signal Cont2 from the CPU 13 (i.e., information relating to object display), and regeneration data Rg corresponding to the scene 120 composed of plural objects is output.

The decoding operation mentioned above is similar to that of the conventional object data decoding apparatus.

The object data decoding apparatus 101 according to this first embodiment is characterized by that the object table T1 shown in figure 2 is created by the CPU 13.

Hereinafter, a description is given of the object table creation process. Figure 3 is a flowchart showing the algorithm for creating the object table T1 by the CPU 13.

Initially, in step S1, the composition stream is read into the data storage of the CPU 13. In step S2, the stream association table is read into the data storage of the CPU 13. In step S3, descriptors of the respective objects on the composition table are loaded into a processor of the CPU 13 wherein an object id value Oid is given to each descriptor, whereby each object can be identified by the object id.

In step S4, it is decided whether or not the object of which descriptor has been loaded corresponds to a node and has no stream. When it corresponds to a node and has no stream, in step S11, the layer of object of which descriptor is to be loaded is lowered by one, followed by step S3 wherein the descriptors of the lower objects being components of the object corresponding to the node are loaded into the CPU 13.

When it is decided in step S4 that the object of which descriptor has been loaded is not one corresponding to a node and having no stream, in step S6, the object id value Oid is entered as a component of the object table.

Thereafter, in step S7, stream association table is interpreted and, according to the result of the interpretation, various kinds of table components corresponding to each object are entered in the object table. The main table components are as follows: the logical channels (LC) corresponding to the respective objects, the priority order of the respective objects, the stream indices (id) corresponding to the respective objects, and the stream type (i.e., Video or Audio). Besides, the following table components are also entered: the indices of the upper and lower objects corresponding to the respective objects, the logical channels of the upper and lower objects, and the indices of objects which share their logical channels with other objects.

In step S8, it is decided whether entry of table components relating to objects that belong to the same node as an object currently being processed by the CPU 13 has been completed or not. When it has not been completed, the control of the CPU 13 returns to step S3, followed by steps S4 to S8. On the other hand, when it is decided in step S8 that entry of table components has been completed with respect to all the objects of the node to which the object currently being processed belongs, the CPU control proceeds to step S9 wherein it is decided whether or not the object currently being processed is the uppermost-layer object in the hierarchy.

When the object currently being processed is not the uppermost-layer object, a process of raising the object layer by one is carried out in step S12, followed by the decision in step S9. On the other hand, when it is decided in step S9 that the object currently being proc-

essed is the uppermost-layer object, the CUP control proceeds to step S10 wherein it is decided whether entry of table components has been completed or not with respect to all the objects constituting one scene.

When it is decided that entry of table components of all the objects has not been completed yet, the CPU control returns to step S3, followed by steps S4 to S9, S11 and S12. On the other hand, when the decision in step 10 is that entry of table components of all the objects has been completed, the object table creation process by the CPU 13 is ended.

In the object data decoding apparatus 101 according to this first embodiment, the object table so created is stored in the data storage of the CPU 13. The stored object table will be updated at every updating of the composition stream and the stream association table so that it can correspond to the updated information. Accordingly, the object table is updated only when any of the objects constituting one scene is changed.

When the multiplexed bit stream includes a flag showing that the updated composition stream and stream association table are sent, the object table may be updated only when the flag is newly sent.

When the object data decoding apparatus 101 performs decoding to a specific object according to a request from the user, a logical channel LC corresponding to the specific object is specified on the basis of the object table stored in the data storage of the CPU 13, and only a packet having this logical channel LC is extracted from the multiplexed bit stream for decoding.

For example, when only the object 122 (Oid=2) corresponding to a node is subjected to decoding, in the multiplexed bit stream of the data structure according to the prior art, it is necessary to interpret the composition stream and the stream association table by the CPU and specify the logical channel LC corresponding to the object 122. In this first embodiment, however, since the object table shown in figure 2 is included in the multiplexed bit stream MEg, the logical channel LC corresponding to the object 122 (i.e., OLC=6~9) can be specified in a moment, resulting in high-speed retrieval.

Further, when the throughput of the decoding apparatus is low and it cannot decode all the objects, it is considered to decode only objects having high priorities. In this first embodiment, since the object table contains the priority order of the respective objects, it is easy to specify the logical channels of high-priority objects.

Furthermore, since the object table contains the indices of objects that share a logical channel with other objects, the following effect is expected.

That is, in the bit stream according to this embodiment, the objects 124 and 125 having Oid=4 and Oid=5 as their object ids (see figure 18) share coded data of the same logical channel LC (OLC=6). So, although these objects have different object id values and different stream id values, these objects have the same value (OLC=6) of corresponding logical channel LC.

When coded data of a specific object is deleted from the multiplexed bit stream, according to the object id of an object relating to the specific object, a logical channel LC corresponding to the relevant object is decided and, thereafter, coded data of the specific object is extracted from the multiplexed bit stream. However, when the specific object shares its logical channel with the relevant object, if coded data of the specific object is deleted, coded data corresponding to the relevant object is gone, whereby decoding of the relevant object cannot be carried out.

In this first embodiment, since the object table T1 contains the index of object that shares its logical channel with another object, this index can be used for deciding whether coded data of the object can be deleted or not, whereby the above-mentioned problem is avoided.

As described above, in this first embodiment of the invention, on the basis of the multiplexed bit stream including coded data Eg corresponding to plural objects, the object data showing the correspondences between the respective objects and the coded data is created in advance and, using the object table, extraction, selection, or retrieval of a specific object is carried out. Therefore, as compared with the case where retrieval or the like of a specific object is carried out by interpreting information about the object included in the multiplexed bit stream each time, the processing quantity required for retrieval or the like is reduced, resulting in high-speed processing.

Further, on the object table T1, since the correlation of plural objects constituting one scene and having a hierarchical structure is described clearly, even when plural objects are included in one object, replacement and edition of the objects are facilitated.

Although in this first embodiment the objects that share a logical channel are shown by their object ids, flags such as "1" and "0" may be used to show only whether or not an object shares its logical channel with another object is required. In this case, the object table is simplified.

Further, the object table is not restricted to that shown in figure 2. Hereinafter, a description is given of an object data decoding apparatus which creates an object table different from the object table shown in figure 2, according to a modification of the first embodiment.

Figure 4 is a diagram for explaining an object table created by the decoding apparatus according to the modification, wherein a table corresponding to upper-layer objects and a table corresponding to lower-layer objects are illustrated.

The object table T2 shown in figure 4 is different from the object table T1 shown in figure 2 in that the table T2 does not have logical channels (LC) of the respective objects, and the table T2 is divided into two parts, i.e., an upper-layer table T2a and a lower-layer table T2b.

The process of creating the object table T2 is differ-

ent from the process flow shown in figure 3 only in that the logical channels (LC) are not entered as table components. So, the object table T2 can be created according to a process flow similar to the process flow shown in figure 3.

By the way, as mentioned above, the bit stream is not always one in which coded data corresponding to the respective objects are multiplexed. In a bit stream in which coded data are not multiplexed, no stream association table is included. So, no logical channel LC is obtained from this bit stream.

In such a bit stream, the logical channels in the multiplexed bit stream correspond to streams, so that coded data of each object is specified using the stream id instead of the logical channel LC.

Therefore, on the object table T2 shown in figure 4, like the object table T1 shown in figure 2, an object index (id) for identifying each object is entered to make the relationship between the object and the stream clear.

Further, on the object table T2 shown in figure 4, in order to make the object hierarchy clear, "H" is adopted as a code showing hierarchical information and described in the section of the kind of stream. With respect to the object having "H" in this section, the object table T2b (lower-layer table) corresponding to lower-layer objects included in this object is created.

As described above, since the object table has a hierarchical structure, even when the hierarchy of objects constituting one scene increases, the size of the object table for each layer (T2a or T2b), which is a component of the object table (T2), does not increase. Therefore, when performing edition or replacement to the upper-layer objects, only the upper-layer object table (T2a) of which size is reduced is retrieved, whereby detection of objects is facilitated.

Although in the first embodiment and its modification, object tables are obtained from the multiplexed bit stream shown in figure 3 and the non-multiplexed bit stream shown in figure 4, respectively, object tables are not restricted thereto.

For example, in order to make the object table shown in figure 2 compact, from the table components shown in figure 2, the upper object id, the upper object LC, the priority order, the kind of stream, and the common object id may be deleted.

To the contrary, although the size of the object table is somewhat increased, in order to simplify the operation such as edition or replacement of objects, the logical channel of the composition stream itself and the logical channel of the stream association table itself may be described on the object table, or header information of streams corresponding to video and audio objects may be added to the table.

As described above, the object table created by the object data decoding apparatus according to the present invention may have any structure as long as, on the table, the respective objects are correlated with the stream indices or logical channels of the objects.

Furthermore, in the object coding method, there is a case where the multiplexing relationship is expressed by only the stream association table in the bit stream without including the composition stream in the bit stream, in order to simplify the structure of the decoding apparatus.

In this case, although the object-to-stream correspondence is not uniquely defined, an object table excluding some of table components of the object table T1 shown in figure 2, such as the object id, the stream id, the kind of stream, the priority order, and the common object id, may be created from the stream association table. In this case, the hierarchical relationship of the logical channels corresponding to coded data of the respective objects is clearly defined. Further, an object table excluding some of table components of the object table T2 shown in figure 4, such as the object id, the stream id, the kind of stream, and the priority order, may be created from the stream association table.

[Embodiment 2]

Figure 5 is a block diagram for explaining an object data selecting apparatus as an object data processing apparatus according to a second embodiment of the present invention.

In figure 5, an object data selecting apparatus 102 according to this second embodiment selects and extracts coded data Sg of a specific object from a multiplexed bit stream MEg, according to an instruction signal Iu corresponding to user's instruction. The multiplexed bit stream MEg is identical to that shown in figure 19 wherein coded data of plural objects are multiplexed in units of the objects.

The object data selecting apparatus 102 includes a multiplexed stream interpreter 61, an object selector 62, and a buffer 64. The multiplexed stream interpreter 61 detects a composition stream and a stream association table which are auxiliary data Dsub included in the multiplexed bit stream MEg. The object selector 62 selects and extracts coded data corresponding to a specific object from the multiplexed bit stream MEg according to a control signal Cont. The buffer 64 is disposed between the multiplexed stream interpreter 61 and the object selector 62, and retains the multiplexed bit stream for a prescribed period of time.

Further, the selecting apparatus 102 includes a CPU 63. The CPU 63 creates the object table T1 shown in figure 2 on the basis of the composition stream and the stream association table, and outputs a signal for selecting coded data of a specified object (control signal Cont) toward the object selector 62 according to an object specifying signal Iu generated as a result of user's operation. Since coded data of the specific object is extracted from the multiplexed bit stream MEg, the contents described in the composition stream and the stream association table change. So, the CPU 63

rewrites the auxiliary data relating to the composition stream and the stream association table so that the stream and the table correspond to the extracted object, and outputs the data to the object selector 62. The object selector 62 adds the composition stream and the stream association table, which have been rewritten by the CPU 13, to the coded data of the extracted object when outputting the coded data.

A description is given of the operation of the object data selecting apparatus 102 using a flowchart shown in figure 6.

Initially, in step S71, the object table T1 is created according to the multiplexed bit stream MEg input to the object data selecting apparatus 102.

To be specific, the multiplexed bit stream is interpreted by the multiplexed stream interpreter 61, and the composition stream and the stream association table which are auxiliary data Dsub included in the multiplexed bit stream MEg are detected and supplied to the CPU 63. In the CPU 63, the object table T1 is created according to the auxiliary data Dsub and stored in the data storage. The process of creating the object table T1 is identical to that already described for the first embodiment.

When a signal Iu specifying an object is input by the user or the like (step S72), the CPU 63 retrieves the object table according to the object specifying signal Iu, specifies the logical channel LC of the specified object, and sends the logical channel LC to the object selector 62 as the control signal Cont (step S73).

Subsequently, the CPU 63 rewrites the composition stream (step S74) and rewrites the stream association table (step S75) so that the stream and the table correspond to coded data of the specified object.

To rewrite the composition stream and the stream association table is necessary because the correlation of objects included in the multiplexed bit stream changes between the input multiplexed bit stream and the multiplexed bit stream corresponding to the extracted object. While the rewriting is carried out, the multiplexed bit stream output from the multiplexed stream interpreter 61 is stored in the buffer 64.

When coded data corresponding to each object included in the multiplexed bit stream is input through the buffer 64 to the object selector 62 (step S76), the object selector 62 decides whether the input coded data corresponds to the logical channel LC of the specified object or not, according to the control signal Cont corresponding to the object specifying signal from the CPU 63 (step S77).

As the result of this decision, when the input coded data does not correspond to the specified object, the next coded data is input to the object selector 62 (step S76). On the other hand, when the input coded corresponds to the specified object, the coded data is output as coded data corresponding to the specified object (step S78). The object selector 62 outputs the coded data of the selected object together with the rewritten composition stream and stream association table.

Thereafter, the object selector 62 decides whether or not the output coded data is the last coded data of the specified object. As the result of this decision, when the output coded data is not the last one, above-mentioned steps S76~S79 are repeated. On the other hand, when the output coded data is the last one, it is decided whether output of coded data of all the specified objects is completed or not (step S80). When it is not completed yet, above-mentioned steps S76~S80 are repeated. On the other hand, when it is completed, the process of selecting coded data of specified objects is ended.

When the object selector 62 outputs coded data of a specified object as described above, the transfer rate of the multiplexed bit stream output from the object data selecting apparatus 102 is lowered as compared with the transfer rate of the multiplexed bit stream input to this apparatus. So, the object selector 62 can change the transfer rate as desired. However, the transfer rate may be changed by independent means located on the output side of the selector 62.

As described above, according to the second embodiment of the invention, on the basis of a bit stream in which coded data Eg corresponding to plural objects are multiplexed, an object table on which the respective objects are correlated with the coded data thereof is created in advance. Using the object table, coded data corresponding to a specific object is extracted from the bit stream. Therefore, it is possible to extract or delete coded data of a specific object from the multiplexed bit stream at high speed in the middle of a transmission path or the like.

Although in this second embodiment a specific object extracted is transmitted, a specific object extracted may be deleted to transmit the rest of the stream.

[Embodiment 3]

Figure 7 is a block diagram for explaining an object data recording apparatus according to a second embodiment of the present invention.

In figure 7, an object data recording apparatus 103 includes a data storage 84, and an object data selecting unit 8 that selects and extracts data stored in the data storage 84. The recording apparatus 103 records the multiplexed bit stream MEg in the data storage 84, and retrieves or outputs coded data of a specified object from the stream stored in the data storage 84 according to user's instruction or the like.

The object data selecting unit 8 includes a multiplexed stream interpreter 81 and an object selector 82. The multiplexed stream interpreter 81 detects a composition stream and a stream association table which are auxiliary data Dsub included in the multiplexed bit stream MEg. The object selector 82 selects and extracts coded data corresponding to a specific object from the multiplexed bit stream MEg according to a con-

trol signal Cont.

Further, the object data selecting unit 8 includes a CPU 83. The CPU 83 creates the object table T1 shown in figure 2 on the basis of the composition stream and the stream association table, and records the object table in a prescribed region of the data storage 84, for example, a region where management information showing the contents of the storage 84 is recorded, or a region where object tables are managed collectively. The object table may be recorded in the same region as where the multiplexed bit stream MEg is stored so that the table is positioned at the head of the stream MEg.

Further, the CPU 83 outputs a signal for selecting coded data of a specified object (control signal Cont) toward the object selector 82 according to an object specifying signal Iu generated as a result of user's operation. Furthermore, the CPU 83 rewrites the auxiliary data relating to the composition stream and the stream association table so that the stream and the table correspond to the selected object, and adds the rewritten data to the coded data of the selected object when the coded data is output.

A description is given of the operation of the object data recording apparatus 103.

Figure 8 is a flowchart showing process steps of creating the object table.

When the multiplexed bit stream MEg is input to the object data recording apparatus 103 and stored in the data storage 84 (step S91), an object table corresponding to the multiplexed bit stream is created in the object data selecting unit 8 controlled by the CPU 83.

More specifically, the multiplexed bit stream is input to the multiplexed stream interpreter 81 (step S92), and the composition stream and the stream association table which are auxiliary data Dsub included in the multiplexed bit stream MEg are detected by the interpreter 81 and supplied to the CPU 83. In the CPU 83, an object table (refer to figure 2) is created according to the auxiliary data Dsub (step S93). The object table created by the CPU 83 is stored in the data storage 84 (step S94). Thereafter, it is decided whether an instruction to end the process of selecting object data is input or not. When there is no end instruction, above-mentioned steps S91~S95 are repeated. When the end instruction is input, the object data selecting process is ended.

The object table is stored in a prescribed region of the data storage 84, for example, a region where management information showing the contents of the storage 84 is recorded, a region where object tables are managed collectively, or the same region as where the multiplexed bit stream MEg is stored (in this case, the table is stored at the head of the stream).

When a signal specifying an object is input to the recording apparatus 103, in the CPU 83, the object table stored in the data storage 84 is retrieved, a logical channel LC corresponding to the specified object is specified, and the specified logical channel LC is output to the object selector 82.

In the object selector 82, on the basis of the logical channel LC from the CPU 83, coded data corresponding to the specified logical channel is selected from the multiplexed bit stream, and the selected coded data Se is output. When the composition stream and the stream association table are changed due to the object selection, the CPU 83 rewrites the stream and the table. The rewritten stream and table are input to the object selector 82 wherein the rewritten stream and table are added to the selected object to be output.

As described above, according to the third embodiment of the present invention, in an apparatus for recording a multiplexed bit stream including coded data of plural objects constituting a single image, an object table on which the respective objects are related with the coded data thereof is created on the basis of the multiplexed bit stream, and the coded data and the object table corresponding to the coded data are recorded. Therefore, the recorded multiplexed bit stream is collectively managed by the object table, whereby the process of retrieving and outputting a desired object from the recorded multiplexed bit stream is performed at high speed.

In the object data recording apparatus 103, since the data storage 84 can serve as a buffer, no buffer is disposed between the multiplexed stream interpreter 81 and the object selector 82. However, a buffer as shown in figure 5 may be disposed between the interpreter 81 and the selector 82.

[Embodiment 4]

Figure 9 is a block diagram illustrating an object data multiplex coding apparatus 104a including an object data output unit 104 which is an object data processing apparatus according to a fourth embodiment of the present invention.

The object data multiplex coding apparatus 104a comprises an encoder 87 and the object data output unit 104. The encoder 87 generates coded data corresponding to plural objects constituting one scene by coding data of the respective objects, multiplexes these coded data with a composition stream and a stream association table which are auxiliary data Dsub, and output the multiplexed data. The object data output unit 104 adds an object table on which the respective objects are correlated with coded data of the objects to the multiplexed bit stream MEg output from the encoder 87, and outputs the multiplexed bit stream MEg with the object table.

The object data output unit 104 includes a multiplexed stream interpreter 81 which detects the composition stream and the stream association table which are auxiliary data Dsub included in the multiplexed bit stream MEg according to a control signal Cont1, and a buffer 85 which temporarily stores the multiplexed bit stream MEg that is input to the buffer 85 through the multiplexed stream interpreter 81. Further, the output

unit 104 includes a CPU 83 which forms an object table T1 as shown in figure 2 on the basis of the composition stream and the stream association table, and a multiplexer 86 which adds the object table to the multiplexed bit stream output from the buffer 85 according to a control signal Cont2.

The operation of the object data output unit 104 will be described using a flowchart shown in figure 10.

When the multiplexed bit stream generated by compressive multiplexing in the encoder 87 is input to the object data output unit 104 (step S111), the CPU 83 decides whether it is "start of scene" or "change of objects constituting one scene" (step S112). When the decision is neither of "start" and "change", the multiplexer 86 is controlled by the control signal Cont1 so that the input multiplexed bit stream is output as it is.

On the other hand, when it is "start of scene" or "change of objects constituting one scene", the multiplexed stream interpreter 81 is controlled by the control signal Cont1 so that the multiplexed bit stream MEg is input to the multiplexed stream interpreter 81 and processed (step S113). In the CPU 83, the composition stream is detected from the multiplexed bit stream (step S114) and, subsequently, the stream association table is extracted from the multiplexed bit stream (step S115) and, further, an object table is created on the basis of the composition stream and the stream association table (step S116).

Thereafter, in the CPU 83, at the time of scene start or scene change, the created object table is added at the head of the multiplexed bit stream output from the buffer 85 (step S117), and the multiplexed bit stream with the object table is output (step S118).

Thereafter, it is decided whether an instruction to end the process of outputting the multiplexed bit stream from the encoder 87 is input or not (step S119). When there is the end instruction, the output process is ended. When there is no end instruction, above-mentioned steps S111 to S119 are repeated.

As described above, according to the fourth embodiment of the invention, the object data output unit 104 receives a multiplexed bit stream obtained by multiplexing coded data of plural objects constituting one scene, adds an object table on which the respective objects are correlated with their coded data to the multiplexed bit stream, and outputs the multiplexed bit stream with the object table. Therefore, it is not necessary for an object data decoding apparatus receiving the multiplexed bit stream MEg with the object table to create an object table, whereby an object data decoding apparatus providing the same effects as the decoding apparatus according to the first embodiment can be realized with simplified structure.

Employed as an object data decoding apparatus to which the multiplexed bit stream MEg and the object table are input may be either a decoding apparatus in which the object table and the coded data of the respective objects in the multiplexed bit stream MEg are stored in different storage regions or a decoding apparatus in which the object table and the coded data are stored in the same storage region.

While in this fourth embodiment the object data output unit 104 outputs the multiplexed bit stream after adding the object table at the head of the stream, the structure of the output unit is not restricted thereto. For example, according to the application, the object table may be inserted in the middle of the multiplexed bit stream and, in this case, the capacity of the buffer 85 can be decreased.

Further, in this fourth embodiment, the encoder 87 simply generates coded data corresponding to the respective objects, and the output unit 104 receives the multiplexed bit stream generated by the encoder and outputs the multiplexed bit stream after adding the object table to the stream. However, the structures of the encoder and the output unit are not restricted thereto. For example, the encoder may create the object table simultaneously with formation of the composition stream and the stream association table, and add the object table to the multiplexed bit stream separately from the composition stream and the stream association table when outputting the multiplexed bit stream. Or, the object table may be output as a part of the composition stream. In this case, the multiple stream interpreter 81 in the output unit is dispensed with, and a conventional multiplexer can be used in the output unit.

Although in the second to fourth embodiments a multiplexed bit stream is described as input coded data, these embodiments are not restricted thereto.

For example, an input bit stream may be a bit stream in which coded data are partitioned in units of objects as described for the modification of the first embodiment. Also in this case, the same effects are obtained by creating the object table as shown in figure 4.

Especially, even when the object data output unit 104 according to the fourth embodiment is constructed so that it receives such a non-multiplexed bit stream and creates the object table shown in figure 4, employed as an object data decoding apparatus which receives the multiplexed bit stream MEg with the object table output from the output unit may be either a decoding apparatus in which the object table and the coded data of the respective objects in the multiplexed bit stream MEg are stored in different storage regions or a decoding apparatus in which the object table and the coded data are stored in the same storage region.

Furthermore, although in the first to fourth embodiments a multiplexed bit stream including coded data corresponding to video data and audio data is described, any multiplexed bit stream may be employed as long as it includes coded data of plural objects constituting individual information to be recorded or transmitted. For example, the respective embodiments may employ a multiplexed bit data including, as coded data of objects, only coded data of video data, audio data, or

computer data, or a multiplexed bit data including coded data of other data.

Furthermore, in the first to fourth embodiments, the object table is created on the basis of the composition table and the stream association table which are included in the multiplexed bit stream MEg as auxiliary data. However, in an information transmission system corresponding to MPEG4 which is currently being standardized, the format of scene description is different from that according to the composition table, and an object descriptor showing the correspondence between object id and stream id is employed in place of the stream association table showing the correspondence between stream id and logical channel.

A description is now given of an object data transmission system according to MPEG4.

Figure 11 is a diagram illustrating the structure of the object data transmission system 200.

In this system 200, coded video data Ev and coded audio data Ea corresponding to objects constituting a single scene 201 and system information Si as auxiliary data are multiplexed by a multiplexer 202, and a multiplexed bit stream MEg1 obtained as a result of the multiplication is transmitted through a transmission medium or stored in a storage medium.

The multiplexed bit stream MEg1 transmitted through the transmission medium or read from the storage medium is demultiplexed (divided) into the coded data Ev and Ea, and the system information Si by a demultiplexer 203.

To be specific, the scene 201 is composed of a background object OB1 (scenery), a sound object OB4 attendant on the background object OB1, a foreground object OB2 (person), and a voice object OB3 attendant on the foreground object OB2. The coded video data Ev is divided into coded data Ev1 corresponding to the background object OB1 and coded data Ev2 corresponding to the foreground object OB2. The coded audio data Ea is divided into coded data Ea1 corresponding to the voice object OB3 and coded data Ea2 corresponding to the sound object OB4. The system information Si as auxiliary data is divided into scene description information Sf and object descriptor OD.

Receiving the respective data separated from the multiplexed bit stream MEg1, a decoder 204 generates regeneration data Rg corresponding to the scene 201 according to these data.

That is, in the scene description information Sf, the hierarchical structure of the objects OB1~OB4 is described together with the relationship between the objects in each layer and their object indices. In the object descriptor OD, the relationship between the object indices and the stream indices (i.e., coded data corresponding to the objects) are described.

Accordingly, the decoder 204 performs decoding and composition of coded data of the respective objects on the basis of the scene description information Sf and the object descriptor OD, and generates a regeneration data Rg for displaying the scene 201.

Furthermore, in the system 200 in figure 11, the objects and the object descriptors are defined with the video data being distinguished from the audio data.

[Embodiment 5]

An object data decoding apparatus of a fifth embodiment of the present invention in the object data transmission system 20 will now be described.

Figure 12 is a block diagram showing the object data decoding apparatus of the fifth embodiment. Note that the objects and the object descriptors are defined without distinguishing the video data from the audio data in this embodiment.

Referring to figure 12, an object data decoding apparatus 105 is shown. The object data decoding apparatus 105 is used for receiving a multiplexed bit stream MEg1 comprising a scene description information Sf and an object descriptor OD rather than the composition stream and the stream association table as the auxiliary data, and reproducing video data Rg of one scene from the multiplexed bit stream MEg1.

The multiplexed bit stream MEg1 comprises coded data in which scene data of one scene 150 in figure 13(a)has been coded for each object of the scene, and the auxiliary data.

Referring to figures 13(a) and 13(b), the scene 150 comprises plural objects (small images) of a hierarchical structure. More specifically, the scene 150 comprises a background image 151 as a background, a mobile 152 moving in the background, logo (Let's start) 153 displayed on the background image, and first and second wheels 154 and 155, which correspond to the objects . The background image 151 serves as a node, and the mobile 152 and the logo 153 belong thereto. Also, the mobile 152 serves as a node, and the first and second wheels 154 and 155 belong thereto. Coded data of the mobile 152 comprises coded data of a window 152a, a body 152b, and a chimney 152c.

The auxiliary data of the multiplexed bit stream MEg1 comprises the scene description information and the object descriptor. Figure 14(a) shows a scene description SD1 on the basis of the scene description information.

The scene description SD1 describes the scene 150. In figure 14(a), Object(1) 161 to Object(5) 165 are shown, which are descriptors which indicate the background image 151, the mobile 152, the logo 153, and the first and second wheels 154 and 155, respectively. As is seen from these descriptors, the mobile 152 and the logo 153 belong to the background image 151, and the first and second wheels 154 and 155 belong to the mobile 152. To each of the descriptors 161 to 165, Object id (Oid) by which coded data of respective objects of the multiplexed bit stream MEg1 can be identified is given ("id" indicates index). Specifically, to the descriptors 161 to 165, Object id (Oid) 1 to 5 are given,

respectively.

Furthermore, in an object descriptor OD shown in figure 14(b), correspondence between Object id and Stream id is shown. As shown in figure 14(b), Object id (Oid=1), Object id (Oid=2), Object id (Oid=3), and Object id (Oid=4, 5), correspond to Stream id (Sid=1, 2), Stream id (Sid=3 to 6), Stream id (Sid=7), and Stream id(Sid=8), respectively.
Furthermore, in the object descriptor OD, data type of each object, i.e., "Video" or "Audio" is described.

The object data decoding apparatus 105 of the fifth embodiment will be described hereinafter.

Referring to figure 12 again, a basic construction of the object data decoding apparatus 105 is identical to that of the object data decoding apparatus 101 of the first embodiment. Specifically, the decoding apparatus 105 comprises a demultiplexer 11a for extracting the scene description information Sf and the object descriptor OD as auxiliary data Dsub included in the multiplexed bit stream MEg1, and extracting coded data Eg of each object from the multiplexed bit stream MEg1 in accordance with a first control signal Cont1, an audio and video (AV) decoder 12 for decoding the coded data Eg in accordance with a second control signal Cont2 and outputting reproduced data Rg of each scene, and a CPU 13a for deciding a stream id of the coded data Eg on the basis of the object descriptor OD and supplying the demultiplexer 11a with the first control signal Cont1 on the basis of the decision result, and supplying the AV decoder 12 with information on placement of objects of one scene and information on display start time of each object using the control signal Cont2 on the basis of the scene description information Sf .

In this fifth embodiment, the CPU is used to create an object table indicating correspondence between objects and the corresponding coded data Eg (stream) on the basis of the scene description information Sf and the object descriptor OD, to be stored in a data storage means in the CPU 13a.

Figure 15(a) and 15(b) are diagrams showing an object table T3 of the scene 150.

Referring to these figures, the object table T3 has a hierarchical structure, and comprises a main table T3a indicating a correspondence between objects of the scene 150 and the corresponding streams, and a sub table T3b indicating a correspondence between video or audio of each object and the corresponding stream.

In these tables, various information associated with Object id is entered.

Specifically, the Object id is given to each object descriptor in the scene description in figure 14(a) to have a one-to-one correspondence, and entered in ascending order of value "Oid" of the Object id .

Object indices (Oid = 1-5) are given to the descriptors 161 to 165 of objects 151 to 151, respectively.

In the main table T3a, type of each object (video or audio), and the corresponding stream id are entered. In the main table T3a, indices of upper and lower objects of each object, stream indices of the upper and lower objects, common object indices of objects which share a stream, and priorities of respective objects are also entered.

The upper object is in a higher-order layer than an object, and the lower object is in a lower-order layer than the object.

Specifically, since no upper object of an upper most object, i.e., the object in a first layer L1a in figure 13(a) exists, Oid of the corresponding upper object id and Sid of the corresponding upper stream id are respectively "0". Upper objects of objects in second and third layers L2a and L3a are objects in the first and second layers L1a and L2a, respectively. In case of a lower most object, i.e., an object having no lower object, Oid of the corresponding lower object id and Sid of the corresponding lower stream id are respectively "0".

In case of objects 151 and 152 each comprising plural pieces of video and audio, the corresponding stream types "H" are described in the main table T3a, and stream indices of coded data of video and audio of these objects are described in the sub table T3b.

As can be seen from the object table T3, an object having Oid=1 comprises two objects, since Oid of the corresponding lower object id "2, 3" is described, and an object having Oid=2 comprises two objects, since Oid of the corresponding lower object id "4, 5" is described.

Subsequently, operation of the object data decoding apparatus 105 of the fifth embodiment will now be described.

Referring to figure 12 again, when the multiplexed bit stream MEg1 is input to the object data decoding apparatus 105, the demultiplexer 11a extracts the scene description information SDI and the object descriptor OD as the auxiliary data Dsub from the multiplexed bit stream MEg1 and outputs the Dsub to the CPU 13a.

The CPU 13a recognizes a correspondence between objects of the scene 150 in figure 13(a) and the corresponding stream indices on the basis of the object descriptor OD, and outputs the first control signal Cont1 on the basis of the result to the demultiplexer 11a.

The demultiplexer 11a collects plural paketed streams in the multiplexed bit stream for each object and outputs the resulting stream to the AV decoder 12.

At this time, the CPU 13a extracts information on a display position and display start time of each object from the scene description information SDI and outputs the information to the AV decoder 12 as the second control signal Cont2.

The AV decoder 12 decodes streams of respective objects (a series of coded data Eg) from the demultiplexer 11a, synthesizes decoded data of respective objects in accordance with the second control signal Cont2, and outputs reproduced data Rg of one scene comprising plural objects.

This decoding is identical to that of the prior art object decoding apparatus as already described.

In this fifth embodiment, in addition to the decoding, the CPU 13a creates the object table T3 in figures 15(a) and 15(b).

Hereinafter, creating of the object table T3 following a flowchart in figure 16.

In step S1a, the scene description information Sf is read to the data storage means of the CPU 13a. In step S2a, the object descriptor OD is read to the data storage means of the CPU 13a. In step S3a, descriptors of respective objects of the scene description SD1 on the basis of the scene description information Sf are loaded into an operating unit of the CPU 13a, and Oid of each object id is given to each descriptor, thereby an object can be identified by the corresponding object id.

In step S6a, the Oid is entered in the object table T3 as "id".

In step S7a, the object descriptor OD is interpreted. On the basis of the interpretation result, various table components of respective objects are entered in the object table T3. The components are the stream id, the priority, the stream type (Video or Audio) of each object. At this time, as the table components, indices of the upper and lower objects, stream indices of the upper and lower objects, and the common object indices are also entered in the object table T3.

In step S9a, it is decided whether the object which is being processed is in an upper most layer or not.

When decided in step S9a that the object is not in the upper most layer, in step S12a an object layer in hierarchy is raised by one, and then in step S9a, the decision step is performed again. On the other hand, when decided in step S9a that the object is in the upper most layer, in step S10a, it is decided whether table components of all objects have been entered in the object table T3 or not.

When decided in step S10a that the components have not been entered, the CPU performs step S3a again and the steps S6a, S7a, S9a, and S12a are performed. On the other hand, when decided the components have been entered, the CPU 13a completes creating the object table.

So created object table is stored in the data storage means of the CPU 13a. Each time the scene description information Sf and the object descriptor OD are updated, the stored table is also updated to describe newest information. Therefore, the object table remains unchanged unless an object of one scene is changed.

When there is a flag in a bit stream indicating that new scene description information Sf and the object descriptor OD have been transmitted, the object table may be updated only when the flag is transmitted.

Also in the object data decoding apparatus 105, the same effects as provided in the object data decoding apparatus 101 of the first embodiment are obtained.

Although in the fifth embodiment, the object data decoding apparatus has been described as the object data processing apparatus in the system according to MPEG4, the object data selecting apparatus of the second embodiment, the object data recording apparatus of the third embodiment, the object data output apparatus of the fourth embodiment can respectively create the object table T3 from the scene description information Sf and the object descriptor OD.

In addition, in the object data decoding apparatus which receives the multiplexed bit stream MEg output from the data output apparatus, creating the object table is dispensed with, and therefore the same effects as in the fifth embodiment are obtained with a simple construction.

Although in this fifth embodiment, the objects and the object descriptors are defined without distinguishing the video data from the audio data, they may be defined with the video data being distinguished from the audio data as in the system 200 in figure 11. In this case, since the data type is clearly shown in the scene description, it is not necessary to describe the data type in the object descriptor.

Furthermore, a program which implements constructions of the processing apparatus and the recording apparatus is recorded in a data recording medium such as a floppy disc, whereby processings in the embodiments are carried out in an independent computer system with ease. This is described below.

Figure 17(a) to 17(c) are diagrams showing signal processing of the object data processing apparatus and the object data recording apparatus of the embodiments in a computer system using a floppy disc which stores a program of the signal processing.

Figure 17(a) shows a front appearance and a cross-section of a floppy disc FD, and a floppy disc body D as a recording medium, and Figure 17(b) shows a physical format of the floppy disc body D.

Referring to figures 17(a) and 17(b), the floppy disc body D is stored in a case F, and in a surface thereof, plural tracks Trs are formed concentrically from outer to inner radius thereof, each track being divided into 16 sectors Se in an angle direction. Data of the program is recorded in allocated areas on the floppy disc body D.

Figure 17(c) is a diagram showing a construction with which the program is recorded/reproduced in/from the floppy disc FD. In case of recording the program in the floppy disc FD, data of the program is written thereto through the floppy disc drive FDD from the computer system Cs. In another case of constructing the image transmission method or image decoding apparatus in the computer system Cs using the program in the floppy disc FD, the program is read from the floppy disc FD by means of the floppy disc drive FDD and transferred to the computer system Cs.

Although image processing in the computer system using the floppy disc as the data recording medium has been described, this image processing is implemented using an optical disc. Further, the recording medium is not limited thereto, and IC card, ROM cassette, or the like may be used so long as it can record a program.

Although the data recording medium which stores

the program of transmission or decoding in the embodiments has been described, it may store the multiplexed bit stream MEg or non-multiplexed bit stream in the embodiments. The data storage means of the recording apparatus of the third embodiment may be realized using the data recording medium in figures 17(a) to 17(c).

Claims

1. An object data processing apparatus for decoding N pieces of coded data (N = positive integer) obtained by compressively coding N pieces of object data which constitute individual data to be recorded or transmitted and have a hierarchical structure, for each object data, said apparatus including:

   hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the N pieces of object data, according to the coded data; and
   table creation means for creating, according to the hierarchical information, an object table on which the respective object data are correlated with coded data corresponding to the respective object data.

2. An object data processing apparatus for decoding N pieces of coded data (N = positive integer) obtained by compressively coding scene data corresponding to one scene, for each of N pieces of objects constituting the scene, said apparatus including:

   hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the respective objects constituting the scene, according to the coded data; and
   table creation means for creating, according to the hierarchical information, an object table on which the respective objects are correlated with coded data corresponding to the respective objects.

3. The object data processing apparatus of claim 2 wherein:

   said hierarchical information extraction means is constructed so that it extracts priority information showing the priority order of the respective objects, according to the coded data, in addition to the hierarchical information; and
   said table creation means is constructed so that it creates, according to the hierarchical information and the priority information, an object table on which the respective objects are correlated with coded data corresponding to the respective objects and the priority order of the respective objects are shown.

4. The object data processing apparatus of claim 2 further including:

   identification information detection means for detecting identification information for identifying coded data of a specific object designated, with reference to the object table; and
   decoding means for extracting coded data of the specific object from the N pieces of coded data according to the identification information, and decoding the extracted coded data.

5. An object data processing apparatus for decoding multiplexed data including N pieces of coded data (N = positive integer) obtained by compressively coding scene data corresponding to one scene, for each of N pieces of objects constituting the scene, said apparatus including:

   hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the N pieces of objects constituting the scene, according to information showing the correlation of the respective coded data and included in the multiplexed data; and
   table creation means for creating, according to the hierarchical information, an object table on which the respective objects are correlated with coded data corresponding to the respective objects.

6. The object data processing apparatus of claim 5 wherein:

   said hierarchical information extraction means is constructed so that it extracts priority information showing the priority order of the respective objects, according to the multiplexed data, in addition to the hierarchical information; and
   said table creation means is constructed so that it creates, according to the hierarchical information and the priority information, an object table on which the respective objects are correlated with coded data corresponding to the respective objects and the priority order of the respective objects are shown.

7. The object data processing apparatus of claim 5 further including:

   identification information detection means for detecting identification information for identifying coded data of a specific object designated,

with reference to the object table; and

decoding means for extracting coded data of the specific object from the multiplexed data according to the identification information, and decoding the extracted coded data.

8. An object data processing apparatus for selecting coded data of a specific object data from N pieces of coded data (N = positive integer) obtained by compressively coding N pieces of object data which constitute individual data to be recorded or transmitted and have a hierarchical structure, for each object data, said apparatus including:

hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the N pieces of object data, according to the coded data; and

table creation means for creating, according to the hierarchical information, an object table on which the respective object data are correlated with coded data corresponding to the respective object data;

said apparatus selecting coded data of a specific object data from the N pieces of coded data with reference to the object table and outputting the selected coded data.

9. An object data processing apparatus for selecting coded data of a specific object from N pieces of coded data (N = positive integer) obtained by compressively coding scene data corresponding to one scene, for each of N pieces of objects constituting the scene, said apparatus including:

hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the respective objects constituting the scene, according to the coded data; and

table creation means for creating, according to the hierarchical information, an object table on which the respective objects are correlated with coded data corresponding to the respective objects;

said apparatus selecting coded data of a specific object from the N pieces of coded data with reference to the object table and outputting the selected coded data.

10. An object data processing apparatus for selecting coded data of a specific object from multiplexed data including N pieces of coded data (N = positive integer) obtained by compressively coding scene data corresponding to one scene, for each of N pieces of objects constituting the scene, said apparatus including:

hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the respective objects constituting the scene, according to information showing the correlation of the respective coded data and included in the multiplexed data; and

table creation means for creating, according to the hierarchical information, an object table on which the respective objects are correlated with coded data corresponding to the respective objects;

said apparatus selecting coded data of a specific object from the multiplexed data with reference to the object table and outputting the selected coded data.

11. An object data recording apparatus having a data storage for storing data, and recording N pieces of coded data (N = positive integer) in said data storage, which coded data are obtained by compressively coding N pieces of object data which constitute individual data to be recorded or transmitted and have a hierarchical structure, for each object data, said apparatus including:

hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the respective object data, according to the coded data; and

table creation means for creating, according to the hierarchical information, an object table on which the respective object data are correlated with coded data corresponding to the respective object data;

said apparatus recording the N pieces of coded data and the object table corresponding to these coded data in said data storage.

12. An object data recording apparatus having a data storage for storing data, and recording N pieces of coded data (N = positive integer) in the data storage, which coded data are obtained by compressively coding scene data corresponding to one scene, for each of N pieces of objects constituting the scene, said apparatus including:

hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the respective objects constituting the scene, according to the coded data; and

table creation means for creating, according to the hierarchical information, an object table on which the respective objects are correlated with coded data corresponding to the respective objects;

said apparatus recording the N pieces of coded

data and the object table corresponding to these coded data in said data storage.

13. The object data recording apparatus of claim 12 wherein the object table corresponding to the N pieces of coded data is added to the N pieces of coded data when being recorded.

14. The object data recording apparatus of claim 12 wherein the object table corresponding to the N pieces of coded data is separated from the N pieces of coded data when being recorded.

15. An object data recording apparatus having a data storage for storing data, and recording multiplexed data including N pieces of coded data (N = positive integer) in said data storage, which coded data are obtained by compressively coding scene data corresponding to one scene, for each of N pieces of objects constituting the scene, said apparatus including:

hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the respective objects constituting the scene, according to information showing the correlation of the respective coded data and included in the multiplexed data; and
table creation means for creating, according to the hierarchical information, an object table on which the respective objects are correlated with coded data corresponding to the respective objects;
said apparatus recording the multiplexed data and the object table corresponding to the multiplexed data in said data storage.

16. The object data recording apparatus of claim 15 wherein the object table corresponding to the multiplexed data is added to the multiplexed data when being recorded.

17. The object data recording apparatus of claim 15 wherein the object table corresponding to the multiplexed data is separated from the multiplexed data when being recorded.

18. A data storage medium containing relevant data relating to individual data to be recorded or transmitted, wherein said relevant data includes an object table on which N pieces of object data (N = positive integer) constituting the individual data and having a hierarchical structure are correlated with N pieces of coded data obtained by compressively coding the respective object data.

19. A data storage medium containing relevant data corresponding to one scene, wherein said relevant data includes an object table on which N pieces of coded data (N = positive integer) obtained by compressively coding scene data corresponding to one scene for each of N pieces of objects constituting the scene are correlated with the respective objects.

20. An object data processing apparatus for outputting N pieces of coded data (N = positive integer) obtained by compressively coding N pieces of object data which constitute individual data to be recorded or transmitted and have a hierarchical structure, for each object data, said apparatus including:

hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the N pieces of object data, according to the coded data; and
table creation means for creating, according to the hierarchical information, an object table on which the respective object data are correlated with coded data corresponding to the respective object data;
said apparatus outputting the N pieces of coded data to which the object table corresponding to these coded data is added.

21. An object data processing apparatus for outputting N pieces of coded data (N = positive integer) obtained by compressively coding scene data corresponding to one scene, for each of N pieces of objects constituting the scene, said apparatus including:

hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the respective objects constituting the scene, according to the coded data; and
table creation means for creating, according to the hierarchical information, an object table on which the respective objects are correlated with coded data corresponding to the respective objects;
said apparatus outputting the N pieces of coded data to which the object table corresponding to these coded data is added.

22. An object data processing apparatus for decoding data output from the object data processing apparatus according to claim 21, including:

data separation means for separating the object table from the output data; and
table storage means for storing the separated object table;

wherein decoding of the coded data corresponding to the respective objects is controlled using the information shown in the object table stored in the table storage means.

23. An object data processing apparatus for outputting multiplexed data including N pieces of coded data (N = positive integer) obtained by compressively coding scene data corresponding to one scene, for each of N pieces of objects constituting the scene, said apparatus including:

hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the respective objects constituting the scene, according to information showing the correlation of the respective coded data and included in the multiplexed data; and
table creation means for creating, according to the hierarchical information, an object table on which the respective objects are correlated with coded data corresponding to the respective objects;
said apparatus outputting the multiplexed data to which the object table corresponding to the multiplexed data is added.

24. An object data processing apparatus for decoding data output from the object data processing apparatus according to claim 23, including:

data separation means for separating the object table from the output data; and
table storage means for storing the separated object table;
wherein decoding of the coded data corresponding to the respective objects is controlled using the information shown in the object table stored in the table storage means.

25. A data structure for transmitting N pieces of coded data (N = positive integer) obtained by compressively coding N pieces of object data which constitute individual data to be recorded or transmitted and have a hierarchical structure, for each object data:
wherein a data group comprising the N pieces of coded data includes an object table on which the respective object data are correlated with coded data corresponding to the respective object data.

26. A data structure for transmitting N pieces of coded data (N = positive integer) obtained by compressively coding scene data corresponding to one scene, for each of N pieces of objects constituting the scene:
wherein a data group comprising the N pieces of coded data includes an object table on which the respective objects are correlated with coded data corresponding to the respective objects.

27. An object data processing apparatus for processing multiplexed data including N pieces of coded data (N = positive integer) and being partitioned into plural packets each having a prescribed code quantity, which coded data are obtained by compressively coding N pieces of object data which constitute individual data to be recorded or transmitted and have a hierarchical structure, for each object data, said apparatus including:

hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the N pieces of object data, according to information showing the correlation of the respective coded data and included in the multiplexed data; and
table creation means for creating, according to the hierarchical information, an object table showing the hierarchical relationship of the plural packets constituting the multiplexed data.

28. An object data processing apparatus for processing multiplexed data including N pieces of coded data (N = positive integer) and being partitioned into plural packets each having a prescribed code quantity, which coded data are obtained by compressively coding scene data corresponding to one scene, for each of N pieces of objects constituting the scene, said apparatus including:

hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the respective objects constituting the scene, according to information showing the correlation of the respective coded data included in the multiplexed data; and
table creation means for creating, according to the hierarchical information, an object table showing the hierarchical relationship of the plural packets constituting the multiplexed data.

29. An object data recording apparatus having a data storage for storing data, and recording, in said storage, multiplexed data which includes N pieces of coded data (N = positive integer) and is partitioned into plural packets each packet having a prescribed coded quantity, which coded data are obtained by compressively coding N pieces of object data which constitute individual data to be recorded or transmitted and have a hierarchical structure, for each object data, said apparatus including:

hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the N pieces of object data, according to information showing the correlation of the respective coded data and included in the multiplexed data; and

table creation means for creating, according to the hierarchical information, an object table showing the hierarchical relationship of the plural packets constituting the multiplexed data;
said apparatus recording the multiplexed data and the object table corresponding to the multiplexed data in said data storage.

30. An object data recording apparatus having a data storage for storing data, and recording, in said data storage, multiplexed data which includes N pieces of coded data (N = positive integer) and is partitioned into plural packets each having a prescribed coded quantity, which coded data are obtained by compressively coding scene data constituting one scene, for each of N pieces of objects constituting the scene, said apparatus including:

hierarchical information extraction means for extracting hierarchical information showing the hierarchical relationship of the respective objects constituting the scene, according to information showing the correlation of the respective coded data and included in the multiplexed data; and
table creation means for creating, according to the hierarchical information, an object table showing the hierarchical relationship of the plural packets constituting the multiplexed data;
said apparatus recording the multiplexed data and the object table corresponding to the multiplexed data in said data storage.

Fig.1

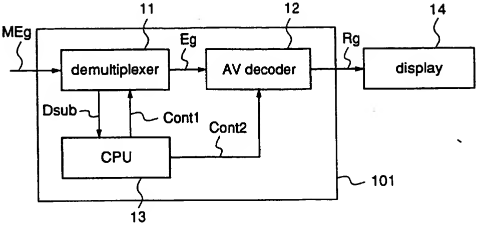

11
12
14
MEg
Eg
Rg
demultiplexer
AV decoder
display
Dsub
Cont1
Cont2
CPU
101
13

Fig.2

object id	LC corresponding to object	stream type	stream id	upper object id	LC corresponding to upper object	lower object id	LC corresponding to lower object	common object id	priority
Oid=1	OLC=3	Video	Sid=1	Oid=1	OLC=3	Oid=0	OLC=0		1
2	4	Video	2	2	4	4,5,6,8	6,7,8,9		1
3	5	Audio	3	3	5	0	0		1
4	6	Video	4	2	4	0	0	Oid=5	5
5	6	Video	5	2	4	0	0	4	5
61	7	Video	6	2	4	0	0		5
62	8	Video	7	2	4	0	0		5
8	9	Audio	8	2	4	0	0		5

T1

Fig.3

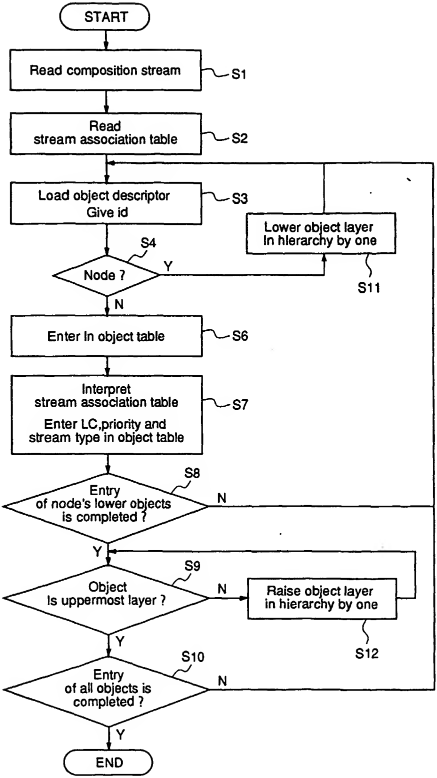

START
Read composition stream
S1
Read
stream association table
S2
Load object descriptor
Give id
S3
Lower object layer
in hierarchy by one
S11
S4
Node ?
Y
N
Enter in object table
S6
Interpret
stream association table
Enter LC,priority and
stream type in object table
S7
S8
Entry
of node's lower objects
is completed ?
N
Y
S9
Object
Is uppermost layer ?
N
Raise object layer
in hierarchy by one
S12
Y
S10
Entry
of all objects is
completed ?
N
Y
END

Fig.4

object id	type	stream id	upper object id	lower object id	priority
Oid=1	Video	Sid=1	Oid=1	0	1
2	Video	2	2	Oid=4,5,6,8	1
3	Audio	3	3	0	1
4	Video	4	2	0	5
5	Video	5	2	0	5
6	H	6, 7	2	Oid=61,62	5
8	Audio	8	2	0	5

T2a
T2b
T2

object id	type	stream id	upper object id	lower object id	priority
Oid=61	Video	6	Oid=2	0	5
62	Video	7	2	0	5

Fig.5

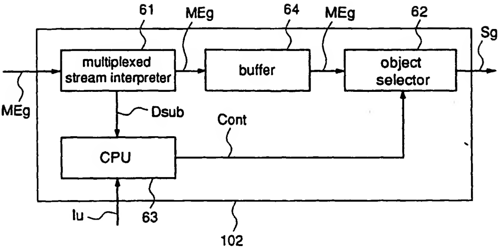

61
MEg
64
MEg
62
Sg
multiplexed stream interpreter
buffer
object selector
MEg
Dsub
Cont
CPU
Iu
63
102

Fig.6

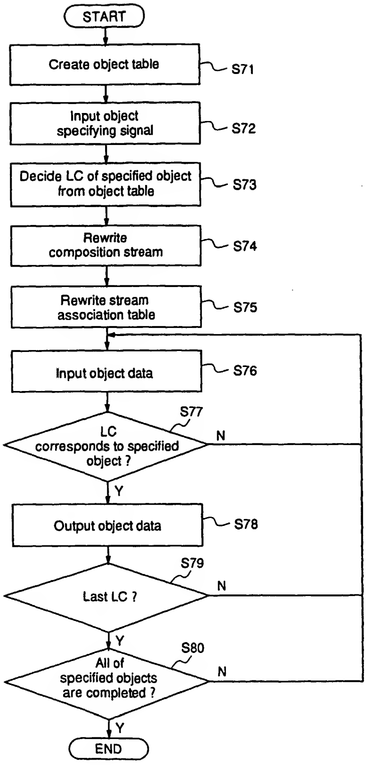

START
Create object table
S71
Input object specifying signal
S72
Decide LC of specified object from object table
S73
Rewrite composition stream
S74
Rewrite stream association table
S75
Input object data
S76
S77
LC corresponds to specified object ?
N
Y
Output object data
S78
S79
Last LC ?
N
Y
S80
All of specified objects are completed ?
N
Y
END

Fig.7

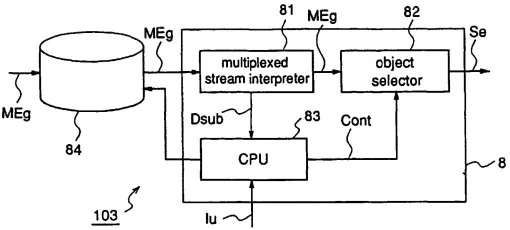

81
MEg
82
Se
MEg
multiplexed
stream interpreter
object
selector
MEg
Dsub
83
Cont
84
CPU
8
103
Iu

Fig.8

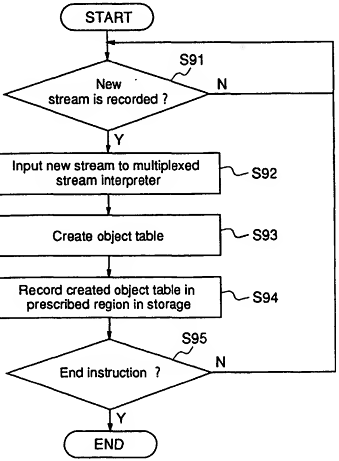

START
S91
New
stream is recorded ?
N
Y
Input new stream to multiplexed
stream interpreter
S92
Create object table
S93
Record created object table in
prescribed region in storage
S94
S95
End instruction ?
N
Y
END

Fig.9

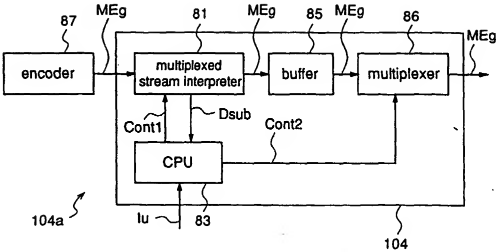
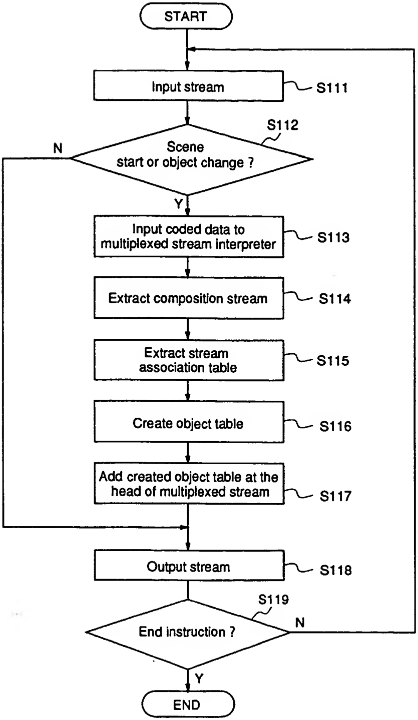

87
MEg
81
MEg
85
MEg
86
MEg
encoder
multiplexed
stream interpreter
buffer
multiplexer
Dsub
Cont1
Cont2
CPU
104a
Iu
83
104

Fig.10


START
Input stream
S111
Scene start or object change ?
S112
N
Y
Input coded data to multiplexed stream interpreter
S113
Extract composition stream
S114
Extract stream association table
S115
Create object table
S116
Add created object table at the head of multiplexed stream
S117
Output stream
S118
End instruction ?
S119
N
Y
END

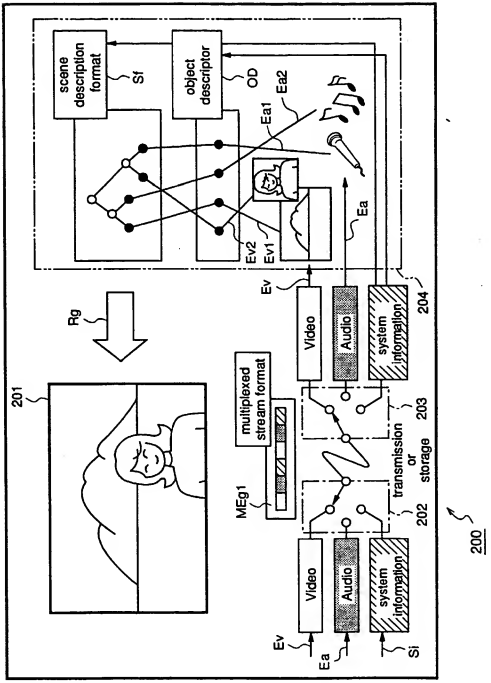

scene description format
Sf
object descriptor
OD
Ea1
Ea2
Ev2
Ev1
Ea
Ev
Video
Audio
system information
204
Rg
201
multiplexed stream format
MEg1
203
transmission or storage
202
Ev
Ea
Si
Video
Audio
system information
200

Fig.11

Fig.12

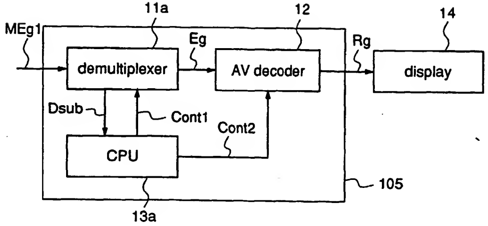

11a
12
14
MEg1
Eg
Rg
demultiplexer
AV decoder
display
Dsub
Cont1
Cont2
CPU
105
13a

Fig.13 (a)

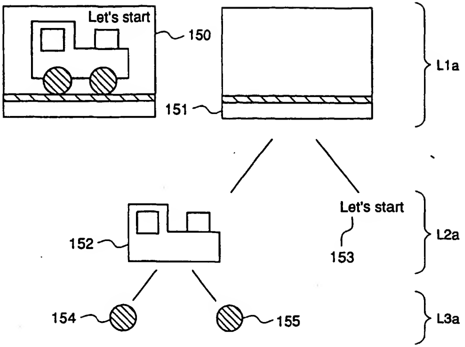


Fig.13 (b)

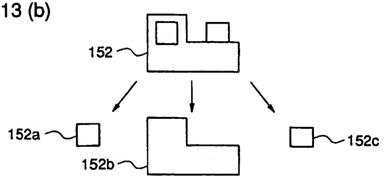


Fig.14 (a)

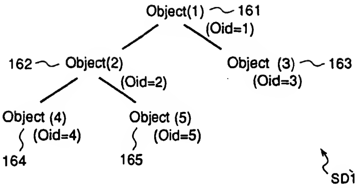


Fig.14 (b)

object id	stream id	data type
Oid=1	Sid=1	Video
1	2	Audio
2	3	Video
2	4	Video
2	5	Video
2	6	Audio
3	7	Video
4	8	Video
5	8	Video

OD

Fig.15 (a)

object id	stream type	stream id	upper object id	upper stream id	lower object id	lower stream id	common object id	priority
1	H	1,2	0	0	2,3	3,4,5,6		1
2	H	3,4,5,6	1	1,2	4,5	8		2
3	Video	7	1	1,2	0	0		3
4	Video	8	2	3,4,5,6	0	0	5	4
5	Video	8	2	3,4,5,6	0	0	4	4

T3a
T3b
T3

Fig.15 (b)

object id	stream type	stream id
1	Video	1
1	Audio	2
2	Video	3
2	Video	4
2	Video	5
2	Audio	6

Fig.16

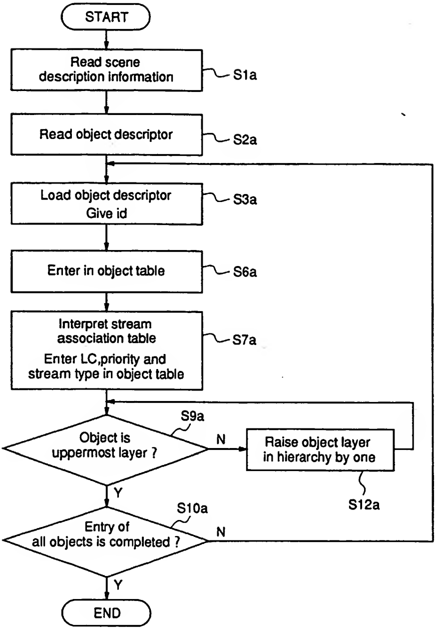

START
Read scene description information
S1a
Read object descriptor
S2a
Load object descriptor
Give id
S3a
Enter in object table
S6a
Interpret stream association table
Enter LC,priority and stream type in object table
S7a
S9a
Object is uppermost layer ?
N
Raise object layer in hierarchy by one
S12a
Y
S10a
Entry of all objects is completed ?
N
Y
END

Fig.17 (a)

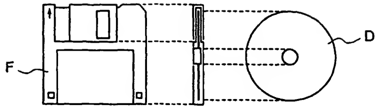


Fig.17 (b)

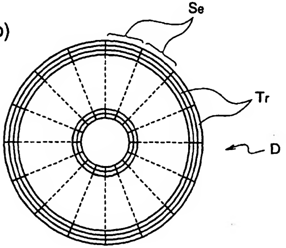


Fig.17 (c)

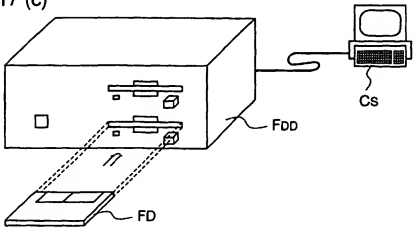


Fig.18 Prior Art

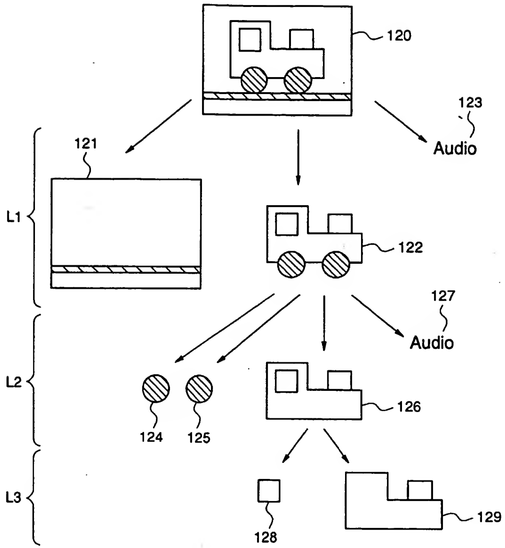


Fig.19 Prior Art

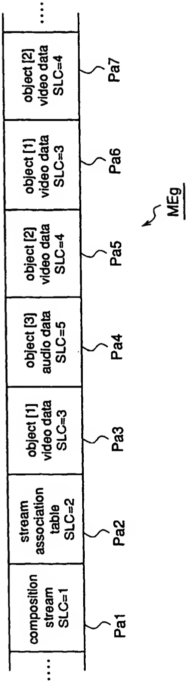


Fig.20 Prior Art

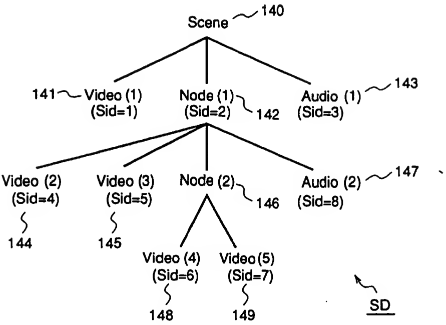


Fig.21 Prior Art

LC corresponding to stream	stream id (Sid)	LC corresponding to upper stream
SLC=3	Sid=1	SLC=2
4	2	2
5	3	2
6	4	4
6	5	4
7	6	4
8	7	4
9	8	4

AT